APPLICATIONS OF NETWORK FLOWS IN LAGRANGIAN RELAXATION

A Thesis Submitted
in Partial Fulfilment of the Requirements
for the Degree of

MASTER OF TECHNOLOGY

by

Sanjeev Swami

to the

DEPARTMENT OF INDUSTRIAL AND MANAGEMENT ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY KANPUR
APRIL, 1994

ACKNOWLEDGMENTS

My research association with Dr. R. K. Ahuja has benefited me in more than one ways. During this period, I have gained guidance and have added value to myself on several fronts - ranging from academic to practical to philosophical and so on. I take this opportunity to express my sincere gratitude to Dr. R. K. Ahuja.

I express my gratitude to all the faculty and staff members of IME family to have made my stay at IITK a memorable one. A special mention in this regard is to be made to Dr. S. Sadagopan, who with his vivacity and ever-accessible guidance, has inspired me *ad infinitum.* I also pay my sincere thanks to the following persons, but for whom this report would not be in this shape - Mr. G. Hoshing at IITK and Mr. P. C. Mishra at 'Dainik Jagaran'.

Finally, I place on record my list of friends, who have kept me amused at 'tight', albeit few, times and have rightly pulled my leg when I used to be on 'overdrive' - Reddy, Yashveer, Captain Ramesh, Sinha, Shah and Arun. I am also thankful to Shiva And Ajay Mishra from senior batches and to Srinivas, Karunesh, Yadav, Devendra, Bansal and others of the enthusiastic lot of 1st year IMEans, for their help and support.

April 12, 1994 Sanjeev Swami

I.I.T. Submitted on 12.4.94

CERTIFICATE

It is certified that the work contained in this thesis entitled **"Applications of Network Flows in Lagrangian Relaxation"** has been carried out under my supervision and that this work has not been submitted elsewhere for a degree.

April 12, 1994.

(Dr. R. K. Ahuja)
Associate Professor
Industrial & Management Engineering
Indian Institute of Technology
Kanpur, INDIA

ABSTRACT

Network flows is an important subfield of Operations Research and is well-known for its diverse applications and highly efficient algorithms. Applications of network flow models have been identified in a variety of situations including applied mathematics, communication systems, defense, manufacturing and production, scheduling, and public policy. In the recent past, researchers have made efforts to identify, compile and summarize those applications which can be transformed into fundamental network flow problems such as shortest paths, maximum flows, minimum cost flows, assignments and minimum spanning trees, etc. This thesis addresses a similar issue and attempts to identify the applications of network flow models in Lagrangian relaxation.

Lagrangian relaxation is an important technique in combinatorial optimization and is extensively used in designing branch and bound algorithms for hard (i.e., NP-complete) problems. The relaxed problems obtained by applying the Lagrangian relaxation technique are often network flow problems and, therefore, can be very efficiently solved by network flow algorithms.

In this work, we make an attempt to identify and summarize such applications whose Lagrangian relaxation results in network flow subproblems. We report over 40 applications which have been selected through an extensive research of the available literature. These also include some applications which we have formulated ourselves. We hope that the understanding of the models described here will allow researchers to identify more and more hard optimization problems whose lower bounds can be obtained more efficiently by using network flow algorithms in Lagrangian relaxation.

CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION

Network flows is an important area in linear programming and is known for its novel theoretical properties. For most of the network flow problems, very efficient algorithms have been developed by exploiting the structure of the constraint matrix. However, many real life and practical problems can be viewed as easy network flow problems complicated by a relatively small set of side constraints. These problems can not be modeled as pure network flow problems and are difficult to solve because direct algorithms are not available for such constrained network flow problems.

Lagrangian relaxation is a flexible solution strategy that permits to exploit underlying special structure in any optimization problem by relaxing complicated constraints. This approach permits to "pull apart" constrained network flow problems by removing complicating non-network constraints and instead place them in the objective function with associated Lagrangian multipliers. Dualizing these complicating side constraints using the Lagrangian relaxation technique produces Lagrangian subproblem (or subproblems) that is a pure network problem in nature. This Lagrangian subproblem can be solved as a pure network flow problem and its optimal value will be a lower bound (for minimization problem) on the optimal value of the original constrained network flow problem. Many practitioners identified various applications of such hard combinatorial optimization problems which have two sets of constraints, one set of "network" constraints and the other of non-network "side" constraints. These applications are few in number and appear to be scattered in the literature.

Though Fisher [1981], and Ahuja, Orlin and Magnanti [1993] gave a compilation of few such applications, yet there is no single paper, or any other reference that summarizes all such real life applications at one place. This report fills precisely this need by describing, summarizing or simply referencing more than 40 such combinatorial optimization problems in which by relaxing complicating non-network constraints, we can get pure network flow Lagrangian subproblems. Furthermore, we also tried to identify and add some applications of similar nature in this report. The network flow problems, which arise as Lagrangian subproblems in such relaxed hard combinatorial optimization problems, are :

(i) The shortest path problem.
(ii) The minimum spanning tree problem.
(iii) The minimum cost flow problem.
(iv) The assignment problem.
(v) The maximum flow problem.

Identifying such constrained network flow application or combinatorial optimization problems with two sets of constraints, proved to be a difficult task. Since there existed no reference on such applications, we almost had to survey the entire combinatorial optimization/network flows area.

This work was started as a project last year on the same topic by Garg [1993] as his M. Tech. thesis. More than 40 applications were collected and reported in that project. This thesis is in continuation of the aforesaid project and collects over 40 applications, which are non-overlapping with those reported in the previous work. In the first phase, we went over the papers cited in the Integer Programming Bibliographies (Kasting [1976] and Hausman [1978]) under the subjects lying within relaxation, network flows and integer programming. We also went through the International Abstracts of Operations Research in above areas. We then scanned these papers to identify such applications. We also scanned the papers given in the reference list of above identified papers. In the second phase, we went through many journals in management science, operations research, mathematics and communications where such articles are likely to have appeared in the past. We also thought and added some additional applications in this phase.

During our research, the applications we found, could be categorized into two broad classes :

(i) Applications whose Lagrangian relaxation resulted into one single pure network flow problem.

(ii) Applications whose relaxation resulted into more than one, different type of pure network flow subproblems.

We have included in our report most of the such applications that were interesting, simple to describe and would throw insight on modeling and relaxing techniques. We have kept the readability and the importance of an application in mind while deciding about its inclusion in the text. There was another category of applications that were nice but either too related to the applications to be described completely or rather too simple to be discussed fully. For such applications, we only

describe the problem. To summarize, the applications identified are covered in this report in the following three manners :

(i) The problem is described and its formulation is shown. By using Lagrangian relaxation, resulting pure network flow subproblem (or subproblems) is shown for such application.

(ii) The problem is described and its equivalence with previously described problem has been proved.

(iii) The problem is simply stated with no description of the model given, because of the complexity of the problem.

1.2 NOTATION AND DEFINITION

We now collect several basic definitions and describe some notations. We consider a directed graph $G = (N, A)$ consisting of a set, N, of nodes, and a set, A, of arcs whose elements are ordered pairs of distinct nodes. A directed network is a directed graph with numerical values attached to its nodes and/or arcs. We let $n = |N|$ and $m = |A|$. We associate with each arc $(i, j) \in A$, a cost c_{ij} and a capacity $u_{ij} \geq 0$. Frequently, we distinguish two special nodes in a graph : the source s and sink t.

An arc (i, j) has two end points, i and j. We refer to node i as the tail and node j as the head of arc (i, j) and the arc (i, j) is incident to nodes i and j. The arc (i, j) is an outgoing arc of node i and an incoming arc of node j. The arc adjacency list of node i, A(i), is defined as the set of arcs emanating from node i, i.e., $A(i) = \{(i, j) \in A: j \in N\}$. The *node-arc incidence matrix* representation, or simply the incidence matrix representation, represents a network as the constraint matrix of the problem. This representation stores the network as an nxm matrix *N* which contains one row for each node of the network and one column for each arc. The column corresponding to arc (i, j) has only two nonzero elements : it has a+1 in the row corresponding to row i and a-1 in the row corresponding to row j. The degree of a node is the number of incoming and outgoing arcs incident to that node.

A directed path in $G = (N, A)$ is a sequence of distinct nodes and arcs $i_1, (i_1, i_2), i_2, (i_2, i_3), i_3,..., (i_{r-1}, i_r), i_r$ satisfying the property that $(i_k, i_{k+1}) \in A$ for $k = 1,...,r-1$. An undirected path is defined similarly except that for any two consecutive nodes i_k and i_{k+1} on the path, the path contains either the arc (i_k, i_{k+1}) or the arc (i_{k+1}, i_k). A directed cycle is a directed path together with the arc (i_r, i_1) and an undirected cycle is an undirected path together with the arc (i_r, i_1) or (i_1, i_r).

We shall often use the terminology *path* to designate either a directed or an undirected path, whichever is appropriate from context. For simplicity of notation, we shall often refer to a path as a sequence of nodes $i_1 - i_2 - \dots - i_k$ when its arcs are apparent from the problem context. Alternatively, we shall sometimes refer to a path as a set of (sequence of) arcs without any mention of the nodes. We shall use similar conventions for representing cycles.

A graph G = (N, A) is called a bipartite graph if its node set N can be partitioned into two subsets N_1 and N_2 so that for each arc (i, j) in A, $i \in N_1$ and $j \in N_2$.

A graph G' = (N', A') is a subgraph of G = (N, A) if $N' \subseteq N$ and $A' \subseteq A$. A graph G' = (N', A') is a spanning subgraph of G = (N, A) if N' = N and $A' \subseteq A$.

Two nodes i and j are said to be connected if the graph contains at least one undirected path from i to j. A graph is said to be connected if all pairs of its nodes are connected; otherwise, it is disconnected. The connected subgraphs of a graph are called its components. We always assume that the graph G is connected and hence $m \geq n-1$.

1.3 NETWORK FLOW PROBLEMS

We shall now briefly describe the network flow problems which arise as Lagrangian subproblems in the relaxed problems considered in this thesis.

Shortest Path Problem

The shortest path problem is to determine directed paths of smallest cost from a given node s to all other nodes in the network. The problem can be stated as the following linear programming problem :

$$\text{Minimize} \quad \sum_{(i\ j) \in A} c_{ij}\, x_{ij}$$

subject to

$$\sum_{\{j:(i\,,\,j) \in A} x_{ij} \;-\; \sum_{\{j:(j,\ i) \in A} x_{ji} = \begin{cases} (n-1) & \text{for } i = s \\ -1 & \text{for } i \in N - \{s\} \end{cases}$$

$$x_{ij} \geq 0 \text{ for all } (i, j) \in A$$

Some of the simplest applications of the shortest path problem are to determine a path between two specified nodes of a network that has a minimum length, or a path that takes least time to traverse, or a path that has the maximum reliability.

Assignment Problem

The data of the assignment problem consists of a set, N_1, say of persons and a set N_2, say of objects, satisfying $|N_1| \equiv |N_2|$, a collection of node pairs $A \subseteq N_1 \times N_2$ representing possible person-to-object assignments, and a cost c_{ij} associated with each element (i, j) in A. The objective is to assign each person to exactly one object in a way that minimizes the total cost of assignment. This problem can be stated as the following linear program:

$$\text{Minimize} \quad \sum_{(i\ j) \in A} c_{ij}\, x_{ij}$$

subject to

$$\sum_{\{j:(i\,,\,j) \in A\}} x_{ij} = 1 \text{ for all } i \in N_1\,,$$

$$\sum_{\{j:(i,\,j) \in A\}} x_{ij} = 1 \text{ for all } j \in N_2,$$

$$x_{ij} \geq 0 \text{ for all } (i, j) \in A.$$

Sometimes, we have a utility u_{ij} assigned with each arc $(i, j) \in A$ and our objective is to obtain an assignment that maximizes the total utility $\sum_{(i\ j) \in A} u_{ij}\, x_{ij}$. We can transform this problem to the previous version by defining $c_{ij} = - u_{ij}$ and minimizing the objective $\sum_{(i\ j) \in A} c_{ij}\, x_{ij}$.

The applications of the assignment problem include assigning people to projects, jobs to machines, tenants to apartments, swimmers to events in a swimming meet, and medical school graduates to available internships.

Minimum Spanning Tree Problem

A spanning tree is a connected acyclic graph that spans all the nodes of an undirected network. The cost of a spanning tree is the sum of the costs of its arcs. In

the minimum spanning tree problem, we wish to identify a spanning tree of minimum cost.

Minimum Cost Flow Problem

We wish to determine a least cost shipment of a commodity through a network that will satisfy demands at certain nodes from available supplies at other nodes. The problem can be stated as the following linear programming program:

$$\text{Minimize} \quad \sum_{(i\ j)\in A} c_{ij}\, x_{ij}$$

subject to

$$\sum_{\{j:(i\ ,\ j)\in A} x_{ij} \ - \sum_{\{j:(j,\ i)\in A} x_{ji} = b(i).$$

$$l_{ij} \le x_{ij} \le u_{ij}.$$

In this problem, c_{ij} represents the cost per unit flow on arc (i, j), l_{ij} represents the lower bound on the arc flow and u_{ij} represents upper bound on arc flow. The number b(i) represents the supply/demand of a node $i \in N$. If $b(i) > 0$, then node i is a *supply* node; if $b(i) < 0$ then node i is a *demand* node; and if $b(i) = 0$ then it is a *transshipment* node.

The applications of minimum cost flow problem include : the distribution of a product from manufacturing plants to warehouses, or from warehouses to retailers; the flow of raw material and intermediate goods through the various machining stations in a production line; the routing of automobiles through an urban street network; and the routing of calls though the telephone system.

Maximum Flow Problem

To define the maximum flow problem, we distinguish two special nodes in the network G: a source node s and a sink node t. We wish to find maximum flow from s to t that satisfies the arc capacities and mass balance constraints at all nodes. The problem is stated formally as follows :

$$\text{Maximize} \quad v$$

subject to

$$\sum_{\{j:(i\,,\,j)\in A} x_{ij} \;-\; \sum_{\{j:(j,\,i)\in A} x_{ji} \;=\; \begin{cases} v & \text{for } i=s \\ 0 & \text{for all } i\in N-\{s\ t\} \\ -v & \text{for } i=t \end{cases}$$

$$0 \le x_{ij} \le u_{ij} \quad \text{for each } (i, j) \in A$$

The applications of the maximum flow problem include : feasible flow problem, matrix rounding problem, scheduling on uniform parallel machines, distributed computing on a two-processor computer, etc.

1.4 OVERVIEW OF THE THESIS

In Chapter 2, we describe the Lagrangian Relaxation technique and related properties and theorems. In Chapter 3, we describe applications in which relaxed Lagrangian subproblems are shortest path problems. Chapter 4 includes applications in which relaxed Lagrangian subproblems are maximum flow problems. Chapter 5 describes the applications in which relaxed problems are minimum cost flow problems. In Chapter 6, we describe applications in which relaxed problems are assignment problems. In Chapter 7, we describe applications whose Lagrangian relaxation results in more than one Lagrangian network flow subproblems. In addition to these, we also report the additional applications in which Lagrangian subproblems will be network flow problems other than the above defined classes. Chapter 8 contains conclusions and direction for future research.

We have mentioned references beside each application. We have not mentioned any reference beside the applications which we have formulated ourselves.

CHAPTER 2

LAGRANGIAN RELAXATION

2.1 INTRODUCTION

Combinatorial optimization problems arise in two situations. There is a small number of easy problems which can be solved in time bounded by a polynomial in the input length and an all-too-large class of "hard problems" for which all known algorithm require exponential time in the worst case. Many hard problems can be viewed as easy problems complicated by a relatively small set of constraints.

Lagrangian relaxation is a solution strategy that dualizes side constraints and produce a Lagrangian subproblem that is easy to solve and whose optimal value is a lower bound (for minimization problem) on the optimal value of the objective function of the original problem. The Lagrangian relaxation can then be used in place of a linear programming relaxation to provide bounds in a branch and bound algorithm.

2.2 LITERATURE SURVEY

The Lagrangian multiplier technique of nonlinear optimization dates back to the 1700s and was originated by the famous mathematician Lagrange, after whom the technique is named. There were a number of forays prior to 1970 into the use of Lagrangian methods in discrete optimization, including the Everett's proposal for "generalizing" Lagrange multipliers (1963). The "birth" of the Lagrangian approach as it exists today, however, occurred in 1970 when Held and Karp (1971) used a Lagrangian problem based on minimum spanning trees to devise a dramatically successful algorithm for the travelling salesman problem. This work was not only an eye-opening successful application, but also set out many key ideas in applying the Lagrangian relaxation method to integer programming problems. Fisher (1981) and Geoffrion (1974) provide insightful surveys of Lagrangian relaxation and its uses in integer programming. Orlin and Bertsimas (1991) have developed the most efficient algorithms (in the worst case) for many classes of Lagrangian relaxation problems. Belling-Seib, Meyer and Muller (1988) have compared Lagrangian relaxation technique for solving the constrained network flow problems with other techniques (primal simplex algorithm and dual method).

2.3 LAGRANGIAN RELAXATION TECHNIQUE

To describe the general form of the Lagrangian relaxation procedure, we begin with the following combinatorial optimization model formulated in terms of a vector X of a decision variables :

$$Z^* = \text{minimize } CX \tag{P}$$

subject to

$$AX \leq b, \tag{2.2a}$$

$$X \in \mathcal{X} \tag{2.2b}$$

The model has a linear objective function CX and a set $AX \leq b$ of explicit linear constrains. The decision variables X are also constrained to lie in a given constraint set $\mathcal{X} = \{X : \mathcal{N}\ X = q, 0 \leq X \leq u\}$, which might be all the feasible solutions to a network flow problem with a supply/demand vector q. We further assume that $\mathcal{X}$ is finite.

Lagrangian relaxation procedure uses the idea of relaxing explicit constraint set (2.2a) by bringing them into the objective function by associating a Lagrangian multiplier vector μ. The resulting problem will be

$$\text{minimize } (CX + \mu\ (AX - b)) \tag{L}$$

subject to $X \in \mathcal{X}$.

We can refer to the function

$$L(\mu) = \min\ \{CX + \mu(AX - b) : X \in \mathcal{X}\}, \tag{2.2c}$$

as the Lagrangian function.

If the constraint set (2.2a) is less than or equal to type of constraint, then μ will be restricted to the non-negative, i.e., $\mu \geq 0$; if it is equality constraint set ($AX = b$), then μ will be unrestricted in sign; otherwise if constraint set (2.2a) is greater than or equal to type of constraint ($AX \geq b$), then μ will be restricted to be negative i.e., $\mu < 0$.

Theorem 2.1 : *(Lagrangian Bounding Principle). For non-negative Lagrangian multipliers $\mu \geq 0$ associated with constraint set (2.2a), i.e., $AX \leq b$, the value $L(\mu)$ of*

the Lagrangian function (2.2c) is a lower bound on the optimal objective function value Z^* of the original optimization problem (P).

Proof : Since $AX \le b$ for every feasible solution to (P) and for non-negative Lagrangian vector set $\mu \ge 0$, the result will be :

$$L(\mu) \le CX^* + \mu(AX^* - b) \le (CX^* = Z^*).$$

We assume that X^* is an optimal solution to (P). As $(AX^* - b)$ will be either zero or less than zero for an optimal solution X^* to (P), for $\mu \ge 0$, $\mu(AX^* - b)$ will be either zero or negative. Hence $L(\mu)$ will be a lower bound on the optimal value Z^* for the original optimization problem. To get the sharpest possible bound, we would need to solve the following optimization problem :

$$L^* = \max_{\mu \ge 0} L(\mu),$$

which we refer to as *Lagrangian multiplier problem* associated with the original optimization problem (P). The Lagrangian bounding principle has the following implications.

Theorem 2.2 : *(Weak Duality Theorem). The optimal objective function L^* of the Lagrange multiplier problem is always a lower bound on the optimal objective function value of the problem (P), i.e., $L^* \le Z^*$.* ♦

Theorem 2.3 : *If for some choice of the Lagrangian multiplier ($\mu \ge 0$), the solution X^* of the Lagrangian relaxation (i) is feasible in the optimization problem (P), and (ii) satisfies the complementary slackness condition $\mu(AX^*-b) = 0$, then X^* is an optimal solution of the original optimization problem (P).*

Proof : By assumption $L(\mu) = CX^* + \mu(AX^* - b)$. Since $\mu(AX^* - b) = 0$, $L(\mu) = CX^*$ and the solution X^* is feasible to the optimization problem (P), i.e., $AX^* \le b$. Hence, X^* is an optimal solution to the optimization problem (P). ♦

The above mentioned properties and theorems show that certain solutions of the Lagrangian subproblems provably solve the original problem (P). From this, we arrive at two properties :

Property 2.1 : Solutions to the Lagrangian subproblem (L) that are feasible, but are not provably optimal for the original problem (P).

Property 2.2 : Solution to the Lagrangian subproblem (L) are not feasible to original problem (P).

In the first case (Property 2.1), branch and bound procedures can be used to get the sharpest possible bound. Fisher (1981) provided an insight about the use of Lagrangian problem in place of linear programming relaxation to provide bounds in branch and bound algorithms. In the second case, for many applications, researchers have been able to modify "modestly" infeasible solutions so that they become feasible with only a slight degradation in the objective function value by using some heuristic methods.

Subgradient Optimization

The following issue is related with the solution procedure of the Lagrangian multiplier problem (L) : How to find out the optimal multiplier value (μ^*) of the Lagrangian multiplier problem (L), i.e., we need to find the highest point of the Lagrangian multiplier function $L(\mu)$.

For our optimization model (P) defined as $\{CX : AX \leq b, X \in X\}$, we assume that the set $X = \{X^1, X^2, \ldots, X^K\}$ is finite. By relaxing the constraints $AX \leq b$, we obtain the Lagrangian multiplier function $L(\mu) = \min \{CX + \mu(AX-b) : X \in X\}$. Thus :

$$L(\mu) \leq CX^k + \mu(AX^k - b) \text{ for all } k = 1,\ldots,K.$$

The best choice for μ would be an optimal solution to the dual problem :

$$\max \omega \qquad \text{(D)}$$

subject to

$$\omega \leq CX^k + \mu(AX^k - b) \text{ for all } k = 1,\ldots,K.$$

$$\mu \geq 0.$$

The above linear program (D) can be solved by applying linear programming methodology. Dantzig-Wolfe decomposition or generalized linear programming is an important solution strategy to solve the linear program (D). It has been discussed by Ahuja, Magnanti and Orlin (1993). However, one of the disadvantages of this approach is that it requires the solution of a series of linear programs which are rather expensive computationally.

Another approach to find the optimal value of Lagrangian multipliers might be to apply gradient method to the Lagrangian function $L(\mu)$. The function $L(\mu)$ has all properties like continuity and concavity except one-differentiability. The function $[L(\mu)]$ is one-differentiable at any μ, where it has two or more solutions. For solving the (non-differentiable) Lagrangian, subgradient optimization technique is commonly used. The subgradient method is a brazen application of the gradient method in which gradients are replaced by subgradients. Let μ^o be any initial choice of the Lagrange multiplier; the subsequent values μ^k for $k = 1,2,...,$ of the Lagrange multipliers can be determined as follow :

$$[\mu^{k+1} = \mu^k + \theta_k (AX^k - b)]^+.$$

In the above expression, X^k is any solution to the Lagrangian subproblem (L) when $\mu = \mu^k$ and θ_k is the step length at the kth iteration. As the updated value of one or more components of μ may become negative, we avoid this possibility by taking only the positive value. The notation $[Y]^+$ denotes the "positive part" of the vector y; that is, the ith component of $[Y]^+$ equals the maximum of zero and Y_i.

Held, Wolfe and Crowder (1974) have discussed the computational performance and theoretical convergence properties of subgradient optimization in detail. The step size θ_k used commonly in practice is :

$$\theta_k = \frac{[UB - L(\mu^k)]}{||(AX^k - b)||^2}$$

In this expression, UB is an upper bound on the optimal objective function value Z^* of the problem (P) and it can be any known feasible solution to the problem (P).

Relationship to Linear Programming

Two properties are important in evaluating a relaxation : the sharpness of the bound produced and the amount of computation required to obtain these bounds. The primary use of the Lagrangian relaxation technique is to obtain lower bounds on the objective function values (discrete) optimization problems. By relaxing the integrality constraints in the original problem (P), we obtain an alternative method for generating lower bound, known as *linear relaxation.*

Theorem 2.3 : *Integrality property : Lagrangian function $L(\mu)$ satisfies the integrality property if it has an integer optimal solution for every choice of μ even if we relax the integrality restriction on the variable X.*

Fisher (1981) has proved that for problems satisfying the integrality property, solving the Lagrangian multiplier problem (L) is equivalent to solving the linear relaxation of the problem (P). In these situations, though Lagrangian relaxation technique provides no better a bound than the linear programming relaxation, yet the Lagrangian relaxation technique might be of considerable value, because solving the Lagrangian multiplier problem (L) might be more efficient than solving the linear programming relaxation directly. It will be an efficient solution strategy where Lagrangian subproblems are network flow problems. The Lagrangian relaxation can exploit the core network substructures - shortest path, minimum cost flow, spanning tree, assignment and other problems. Usually, we choose the constraints to relax in a way that we can exploit underlying structure of the relaxed problem.

Our work is concentrated to identify all such applications in a variety of areas where Lagrangian relaxation of complicating constraints results in network flow subproblem (or subproblems). These Lagrangian subproblems can be solved efficiently by network flow algorithms for a fixed value of Lagrangian multiplier vector and we can get bounds for the optimal solution of the original problem. Ahuja, Magnanti and Orlin (1993) presented a literature survey of network flow problems and described some of the fastest algorithm for solving these network flow problems.

CHAPTER 3

APPLICATIONS OF SHORTEST PATH PROBLEM

In this chapter, we report applications in which the Lagrangian subproblems are shortest path problems. The following is the list of applications reported :

1. Very large scale integrated (VLSI) circuit design.
   (Feo and Hochbaum [1986])
2. Survivable network design (SND) testing
   (Feo and Hochbaum [1986])
3. Crew scheduling in a mass transit setting
   (Cararessi and Gallo [1986])
4. Rostering problem
   (Cararessi and Gallo [1986])
5. Resource constrained shortest path problem
   (Beasley and Christofides [1989])
6. Rotating workforce scheduling
   (Balakrishnan and Wong [1990])
7. Rotating workforce scheduling : Additional applications
   (Balakrishnan and Wong [1990])
8. Traveling salesman problem as a shortest path problem with side constraints
   (Shapiro [1991])
9. Scheduling of power-generation system
   (Muckstadt and Koenig [1977])
10. Multi-item capacitated lot sizing problem
    (Chen and Thizy [1990])
11. Allocation of inspection effort on a production line

Application 3.1 : Very Large Scale Integrated (VLSI) Circuit Design

Reference : Thomas A. Feo and Dorit S. Hochbaum (1986).

The demand for electronic circuits has increased rapidly over the past 50 years. Bell Communications Research, Inc.'s (BELLCORE) interest in VLSI comes from their involvement in telephone and communications systems. In this industry, integrated circuits that perform specialized high-speed digital switching are

of great importance. The method presented in this paper fills the important need for recognizing infeasibility of certain VLSI routing problems.

In VLSI chips, the area or size of chip is a factor of great importance. The goal of design is to permit the manufacturing of a chip that will be as small as possible. For given components (such as arithmetic logic units, memory buffers, and so forth), their placement on chip, and their terminal positions, the design problem is to find within the given area a routing that will connect disjoint set of terminals. This routing must obey certain wiring rules that specify the distances (disjointness) between the routing wires. The disjoint sets of terminals to be connected are called *nets*. For such a design, it needs to be determined whether there is a routing that obeys all the wiring rules and connects all the disjoint pairs of terminals.

According to commonly used wiring models, the physical connections between the terminals must follow paths along a grid. The routing must not coincide along the grid, that is, they are not allowed to share any point on the grid. In the context of graphs, each point on the grid is considered a vertex and connection between two points an edge. Every path from one vertex, a, to another, b, is viewed as an ordered string of vertices, $V_1, V_2, \ldots, V_l$, with $V_1 = a$, $V_l = b$. Given k pairs of terminals, the problem translates into finding k vertex disjoint paths in a grid-like graph. When the graph is any collection of vertices and edges, the problem is known as vertex disjoint path problem (DPP). So the concern is : given a graph G containing k disjoint vertex pairs $(s_1, t_1), (s_2, t_2), \ldots, (s_k, t_k)$, does the graph G contain k vertex disjoint paths, one for each (s, t) pair?

In the formulation, G is considered as a multicommodity flows network with a unique commodity associated with each source-terminal pair. Let x_{ijl} be the flow of commodity l from node i along an edge in E to node j. Let $n = |V|$ and k directed edges be added to G, each from a t_i to s_i for $i = 1,\ldots,k$. Now the DPP formulation is as

$$Z = \max \ \Sigma_{i=1}^{k} \ x_{t_i s_i l}$$

subject to

$$\Sigma_{j=1}^{n} (x_{jil} - x_{ijl}) = 0 \qquad \text{for } i \in V, l = 1,\ldots k \qquad (3.1a)$$

$$\Sigma_{l=1}^{k} \Sigma_{j=1}^{n} x_{jil} \leq 1 \qquad \text{for } i \in V \qquad (3.1b)$$

$$x_{jil} = \{0, 1\} \qquad \text{for } i, j \in V, \ l = 1,\ldots, k \qquad (3.1c)$$

The objective of this formulation is to maximize the sum of flow of k commodities. Constraints (3.1a) stipulate that sum of flows of each commodity into a node must equal the sum of flows of that commodity out of the node. The constraint (3.1b) is a capacity constraint. Constraint (3.1c) is an integrality constraint.

Note that DPP is a recognition problem, and the answer to this problem is yes if and only if $Z^* = k$. Due to the three sets of constraints, every source-terminal pair will possess either a unit flow, representing a simple path in G, or no flow. The objective function value Z^* is the maximum number of vertex disjoint paths between the k pairs of vertices in G. If $Z^* < k$, then G does not contain k vertex disjoint paths and this is the case that is referred to as infeasible. The Lagrangian relaxation presented in this paper follows directly from the previous discussions. The linear formulation of DPP is relaxed by moving the capacity constraints (3.1b) into the objective function. Let π_i be the nonnegative penalty corresponding to the capacity constraints of node i. The Lagrangian relaxation of the vertex disjoint paths problem (RDPP) is

$$\max \Sigma_{i=1}^{k} x_{ti,sil} + \Sigma_{j=1}^{k} \pi_j (1 - \Sigma_{l=1}^{k} \Sigma_{i=1}^{n} x_{ijl})$$

subject to

$$\Sigma_{j=1}^{n} (x_{jil} - x_{ijl}) = 0 \qquad \text{for } i \in V,\ l = 1,...k \qquad (3.1d)$$

$$x_{ijl} = \{0, 1\} \qquad \text{for } i, j \in V,\ l = 1,..., k \qquad (3.1e)$$

Next, we state the following result, as given by the authors : The RDPP can be solved by solving k shortest-path problems.

The proof is given in the paper by Feo and Hochbaum (1986). Thereafter, one can proceed in the usual manner using subgradient optimization technique as described earlier.

Application 3.2 : SND (Survivable network design) Testing

Reference : Thomas A. Feo and Dorit S. Hochbaum (1986).

A possible extension to the recognition version of VLSI routing (as described in application 3.1) is survivable network design (SND) testing. An SND problem of interest to national security is to assess the survivability of a network by the number

of vertex disjoint paths that remain between key vertices after a given damage scenario. Applications of the Lagrangian relaxation method will produce an upper bound for making such an assessment. A similar extension is possible for the problems that arise in communication networks.

Application 3.3 : Crew Scheduling in a Mass Transit Setting

Reference : Carasessi and Gallo (1986).

It consists of finding a set of duties to be assigned to drivers of vehicles in order that the daily service requirements be met at the minimum cost.

It is assumed here that the vehicle scheduling problem has already been solved, and that the set of vehicle duties defining the vehicle schedule is already known. The term "block", as is often used in the literature, is used for "vehicle duty". For each block a set of *relief times* , i.e., times at which a driver's substitution may occur, is given. A *piece of work* is a continuous driving period. Let s(p) and e(p) be the starting and ending time respectively of piece of work p. A piece of work p is feasible for a block if both s(p) and e(p) are relief times of the block. A *duty* consists of a set of pairs (p, k), where p is a piece of work and k is the block on which the piece of work shall be run. Let $T_k = \{t_1^k, t_2^k, ..., t_{u_k}^k\}$ and p_k with $m_k = |p_k|$ denote the set of relief times and the set of pieces of work feasible for block k, respectively; t_1^k and $t_{u_k}^k$ being the starting and ending times of the block, and let $D = \{d_1, d_2, ..., d_{|D|}\}$ be the set of all feasible duties.

Set $\bar{p}_k \subseteq p_k$ is called a partition of block k : a partition of block is a set of pieces of work that cover the service on that block. With the appreciation of partitioning in this way, it becomes apparent that the crew scheduling problem boils down to finding one partition for each block and then matching together pieces of work from the partitions in order to get feasible duties. Out of the many feasible solutions, we seek one which minimizes the total cost.

Associated with each block k, let us define the network $G_k = (N_k, A_k)$, where $N_k = T_k$ and $A_k = \{s(p), e(p) : p \in p_k\}$; i.e. , each node corresponds to one of the relief times, while each arc corresponds to one of the pieces of work feasible for the block; t_1^k and $t_{u_k}^k$ being the origin and destination of the network, respectively.

For instance, consider a block with relief times at 8:30, 9:30, 10:20, 11:20 and 12:30, and assume that feasible pieces of work have lengths between 1 and 2 hours; then the associated network can be shown as in Figure 3.3.

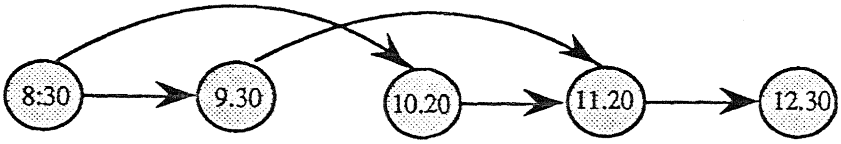


Figure 3.3. An example timing block

A partition of block k corresponds to a path from t_1^k to $t_{u_k}^k$ in G_k and vice-versa. In the given example, the path (8:30, 9:30, 11:20, 12:30) corresponds to partition

$$\bar{p}_k = \{(8{:}30, 9{:}30), (9{:}30, 11{:}20), (11{:}20, 12{:}30)\}$$

Finally, the crew scheduling problem can be formulated as :

[P] Minimize cx

subject to

$$-\sum_{p \in P_k \; e(p)=l} y_p^k + \sum_{p \in P_k \; s(p)=l} y_p^k = \begin{cases} 1 & l = t_1^k \\ 0 & l = t_i^k \;\; i = 2..(u_k\text{-}1) \\ -1 & l = t_{u_k}^k \end{cases} \tag{3.3a}$$

$$\sum_{j \in I_{pk}} x_j = y_p^k, \qquad p \in P_k, k = 1,..r \tag{3.3b}$$

$$Bx \geq a \tag{3.3c}$$

$$y^k \in \{0, 1\}^{mk} \tag{3.3d}$$

$$x \in \{0, 1\}^{|D|} \tag{3.3e}$$

where c - cost vector

I_{pk} - subset of all the duty indices corresponding in G to arcs incident to nodes (p, k)

x - vector of binary variables corresponding to the set of all feasible duties.

$$y = \begin{cases} 1 & \text{if the arc corresponds to a piece of work used in the path} \\ 0 & \text{otherwise} \end{cases}$$

Constraint (3.3c) is added to take into account peculiarities of the various individual problems and matrix 'B' and matrix 'a' have conformable dimensions.

Problem (P) is a rather difficult one, because of the large amount of integer variables and constraints, even for small size transportation companies. The Lagrangian relaxation of [P] is obtained by multiplying constraints (3.3b) and (3.3c) by suitable multipliers and bringing them into the objective function. We get

$$(P_R) \quad L(\lambda, \mu) = \min \left[cx - \Sigma_k \Sigma_p \lambda_{pk}\left(\sum_{j \in I_{pk}} x_j - y_p^{\kappa}\right) - \mu(Bx-a)\right]$$

subject to

(3.3a), (3.3d) and (3.3e) ,

where λ and μ are non-negative multiplier vectors.

It is assumed that constraints (3.3b) was transformed into $\geq$ inequality form. The relaxed problem is a shortest-path problem with modified costs and $L(\lambda, \mu)$ provides a lower bound to the optimal value of problem (P). The best lower bound can be obtained by solving the generalized dual

$$\text{Max}_{\lambda,\mu \geq 0} \; L(\lambda, \mu)$$

which can be done by means of subgradient optimization technique, as described earlier.

Application 3.4 : Rostering Problem

Reference : Cararessi and Gallo (1986)

Once a feasible crew schedule has been obtained, a further problem remains to be solved, called the rostering problem, which is to assign the duties to actual drivers. Depending on the union contract clauses, this can be either a trivial problem or a rather difficult one. In most North American Companies, the assignment of duties is left to the drivers themselves and the decision is made on the basis of seniority, leaving little or no space at all for mathematical modeling. On the other hand, there are cases in which the union contract is so binding that in order to computerize the rostering procedures, one has to resort to rather sophisticated mathematical techniques. This is the case with many European companies, where, other things being equal, the objective is set to distribute the work load evenly among the drivers. The authors address the last type of problem and propose that network models of vehicle scheduling as modelled in previous application extend to these problems as well.

Application 3.5 : Resource Constrained Shortest Path Problem

Reference : Beasley and Christofides (1989)

It is the problem of finding the shortest path between two vertices of a network whenever the traversal of an arc/vertex consumes several resources and the resources thus consumed must lie within given limits. The problem can be linked to that of a traveller with a budget who has to reach a given destination as quickly as possible within the constrains imposed by his budget. The two basic type of problems addressed are :

1. the resource constrained shortest path (RCSP) problem as defined above; and

2. the vertex constrained shortest path (VCSP) problem, which is the problem of finding the shortest path constrained to pass through some specified vertices.

Note that VCSP can be viewed as a special case of RCSP, i.e., simply associate a different resource with each vertex in the set of specified vertices and impose appropriate resource limits. Problem formulation is as:

Consider a network $G = (V, A)$, where V is set of a vertices and A is set of arcs. c_{ij} is length of the arc (i, j) and is infinity if $(i, j) \notin A$ and $c_{ii} = 0 \ \forall \ i \in V$.

Let $x_{ij} = \begin{cases} 1 & \text{if arc (i j) is in the optimal path} \\ 0 & \text{otherwise} \end{cases}$

r_{ijk} = amount of resource k used in traversing arc (i, j)

q_{ik} = amount of resource k used in traversing through vertex i

K = type of resources

The model is as :

[P] $\quad \min \Sigma_{i\in V}\Sigma_{j\in V} \ c_{ij}\, x_{ij}$

subject to

$$L_k \le \Sigma_{i\in V}\Sigma_{j\in V} \ r_{ijk}\, x_{ij} + \Sigma_{j\in V} \ q_{jk} \left(\Sigma_{i\in V} x_{ij} + \Sigma_{m\in V} x_{jm} \right)/2 \le U_k$$

$$\forall \ k = 1 .. \ K \qquad (3.5a)$$

$$\Sigma_{i\in V} x_{ij} = \Sigma_{i\in V} x_{ii} \qquad \forall j \in V \quad (j \ne 1, n) \qquad (3.5b)$$

$$\Sigma_{j\in V} x_{ij} = 1 \qquad (3.5c)$$

$$\Sigma_{i\in V} x_{in} = 1 \qquad (3.5d)$$

$$x_{ij} = (0, 1) \qquad \forall\, i, j \in V \qquad (3.5e)$$

Constraint (3.5a) ensures that the total amount of any resource k used on the path should be between L_k and U_k, . Constraint (3.5b) is the degree constraint for each node in the network except for origin and destination vertices, while constraints (3.5c) and (3.5d) ensure that one arc leaves the origin and one arc enters the destination. The Lagrangian relaxation to generate effective lower bounds for the above formulation is proposed by relaxing constraints (3.5a) and associating non-negative multipliers s_k and t_k with its left-hand and right-hand sides, respectively. The relaxed problem P_R is as :

$$[P_R] \quad \min \Sigma_{i\in V}\Sigma_{j\in V} \left(c_{ij} + \Sigma_{k=1}^{K} (t_k - s_k)\,(r_{ijk} + (q_{ik} + q_{jk})/2)\right) x_{ij}$$

$$+ \Sigma_{k=1}^{K} (s_k L_k - t_k U_k)$$

subject to

(3.5b), (3.5c), (3.5d) and (3.5e)

This program is a shortest path problem., the problem of sending a flow of value one from vertex to all others subject to flow conservation constraint with a capacity limit of one on each arc and the cost of using each arc as the multiplier of x_{ij} in the objective function shown above.

Application 3.6 : Rotating Workforce Scheduling

Reference : Balakrishnan and Wong (1990).

This problem involves the construction of an efficient sequence of work and rest periods spanning over a number of weeks. Typical constraints in the scheduling process impose upper and lower bounds on the lengths of rest and work periods, permit only specific shift changes, and restrict the number of consecutive work periods on the same shift. The schedule is so designed that at the end of the every cycle of M weeks, where M is the planning horizon, the entire schedule will have repeated itself for every worker. The workers would begin on different weeks

of the sequence, i.e., what worker "i" did in week 1, worker "(i-1)" will do in week 2, and so on. Therefore, with this approach, at the end of each cycle, the workload is uniformly distributed between the workers and they all will have equally desirable schedules.

In addition to the regular constraints such as lower and upper bounds on the length of rest periods, etc, there are complicating staff covering constraints for the minimum workforce requirement for each shift of each day of the week. Thus, there are 7 S such constraints where S is the number of shifts in a day. These constraints are relaxed in a Lagrangian fashion in the solution technique.

A basic network model could be associated with this problem in the following way.

The first day in the planning horizon corresponds to the source node in the network, and similarly the last day corresponds to the sink node. The arcs in the network corresponds to a work period allocated to a specific shift. The network is constructed in a way that each path from source to sink will define an alternating sequence of work and rest periods. If the length of the path is modeled as the cost of that sequence or schedule, then the shortest path gives the best sequence or schedule, in terms of cost. An example network for a 4-week, two-shift where the work-period length must be between 3 and 7 days, and the rest-period lengths must be between 1 and 3 days, is as shown in the Figure 3.6.

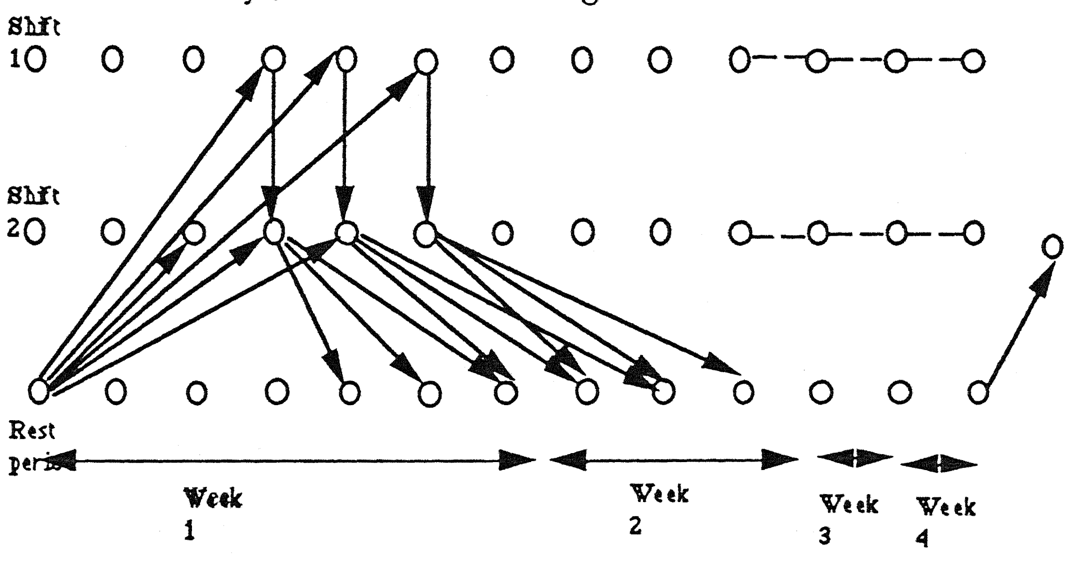


Fig. 3.6. Network for rotating workforce scheduling.

Note that a series of nodes over each day of the planning is shown for shifts as well as rest period. Also, each arc terminating at the shift 1 node define shift 1

work period and similar interpretation is for other arcs. Since a rest period shoul(be there between any two work periods, any arc originating from first row or seconc row terminates in the third row. Dummy arcs have been created between shift 1 and shift 2 nodes so that a rest period can follow shift 1 work period also Alternatively, these can be eliminated and the arcs can be shown to terminatε directly from first to third row.

Since there is selective approach for arcs to be included in the network (based on various constraints), the network is very sparse. The staffing constraints still remain to be taken care of. This is done in the following way. With each arc, associate a vector of attributes that will have as many elements as there are temporal (staffing) requirements. For a 2-shift problem, there will be 14 such elements. For example, an arc indicating a work period from Monday to Friday in the first shift will have the vector as (11111 00 0000000). Each arc corresponding to a rest period has an associated vector of all zeros. The path between the source and the sink should then be such that the attribute vector, summed over all arcs on which flow occurs, is greater than or equal to temporal workforce requirements. This is due to the fact that the planning horizon is of as many weeks as is the total number of workers to facilitate perfect rotation of load. The solution strategy is as follows. To find the optimal solution satisfying all the constraints (including temporal workforce requirements), it is necessary only to find the shortest path through the network that satisfies the temporal constraints. However, the complete enumeration of all the paths starting from the shortest path makes it seem unpractical. Thus, a two-phase approach is proposed. The first phase is to relax the temporal constraints in the Lagrangian fashion and obtain a good lower bound by solving the resulting shortest-path subproblem. The second phase uses a heuristic algorithm that makes use of the bound generated in the first phase.

Application 3.7 : Rotating Workforce Scheduling : Additional Applications

Reference : Balakrishnan and Wong (1990)

The model and the solution procedure of the Application 3.6 have been applied to the following three real-life example that have been quoted in the literature earlier.

(a) Metropolitan Police Department : Rotating Schedules

(Heller, McWeen and Stenzel (1973))

The problem deals with the Evidene Technician Unit from St. Louis Metropolitan Police Department which has 17 officers who are assigned rotating schedules to staff three nonoverlapping shifts every day of the week. The problem has following constraints (i) rest-period length must be between 2 and 7 days; (ii) work-period length must be between 3 and 8 days; (iii) a shift cannot be assigned to more than 4 consecutive weeks in a row; (iv) shift changes are not allowed after a rest period that includes a Sunday or a Monday or both; (v) the only allowable shift changes are 1 to 3, 2 to 1, and 3 to 2 and, (vi) a matrix of temporal requirement has been provided additionally. The solution technique of Balakrishnan and Wong yields smoother workforce scheduling than the original solution.

(b) Edmonston Police Department : Manpower Scheduling

(Butler (1978))

This is to solve a scheduling problem in the Edmonton Police Department, Canada. There are 9 officers involved, and the temporal requirements call for two officers during each shift of every day, except during the second shift of Thursday, Friday and Saturday when three officers are needed. Other constraints are (i) work-period lengths are between 4 and 7 days; (ii) only shift 1 can precede the rest period preceding a shift 3 work period; (iii) before and after weekend off, only shift 3 and shift 2 work-periods are allowed; (iv) at least two consecutive days must be assigned to the same shift, and (v) no more than two 7-day work periods are allowed and these work-periods should not be consecutive. The solution technique of Application 3.6 was applied successfully to this problem as well with provisions made in the network to accommodate additional constraints.

(c) Laporte et.al.'s Police Manpower Scheduling Problem

(Laporte, Nobert and Biron (1980))

The use of integer programming is illustrated to solve the police manpower scheduling problem. There are three nonoverlapping shifts and the constraints in this problem specify that (i) rest periods should be at least two days off; (ii) work periods must be between 2 and 7 days if the shift is 1 or 2 and between 4 and 7 days if work is done in shift 3; (iii) shift changes can occur only after a day off; (iv) schedules should contains as many weekends as possible; (v) weekends off should be distributed throughout as evenly as possible; (vi) long (short) work-periods should be followed by long (short) rest periods, and (vii) work periods of 7 days length are preferred in shift 3. The temporal requirements specify that two workers are

required during each shift of every day. In this case also, the solution technique of application number 3.6 yielded better solution than the original problem.

Application 3.8 : Traveling Salesman Problem (TSP) as Shortest Path Problem with Side Constraints

Reference : Shapiro (1991)

The traveling salesman problem is easy to state. Starting from his home base, node 0, a salesman wishes to visit each of several cities, represented by nodes 1,....,n, exactly once and return home, doing so at the lowest possible travel cost. Any feasible solution to the problem is known as a tour.

Let c_{ij} be the arc length (cost) of arc (i, j). The arc lengths need not be symmetric, i.e., $c_{ij} \neq c_{ji}$ is allowed. A new node with the label N+1 is introduced which is identical with node 0 but corresponds to terminating there. Thus, the arc costs $c_{i,\,N+1} = c_{i0}$ for all i. A shortest route problem with side constraints represents TSP adequately. As illustrated in accompanying figure, for a seven-city problem, the nodes (1,..,N) are replicated N times. The distance from node i in replicate t to node j in replicate (t+1) is c_{ij}. Therefore, TSP can be expressed as the problem of sending one unit of flow from node 0 to node N+1 in the expanded network so that the distance traveled is minimized (a shortest-route problem), but with the additional constraint that each city must be visited exactly once.

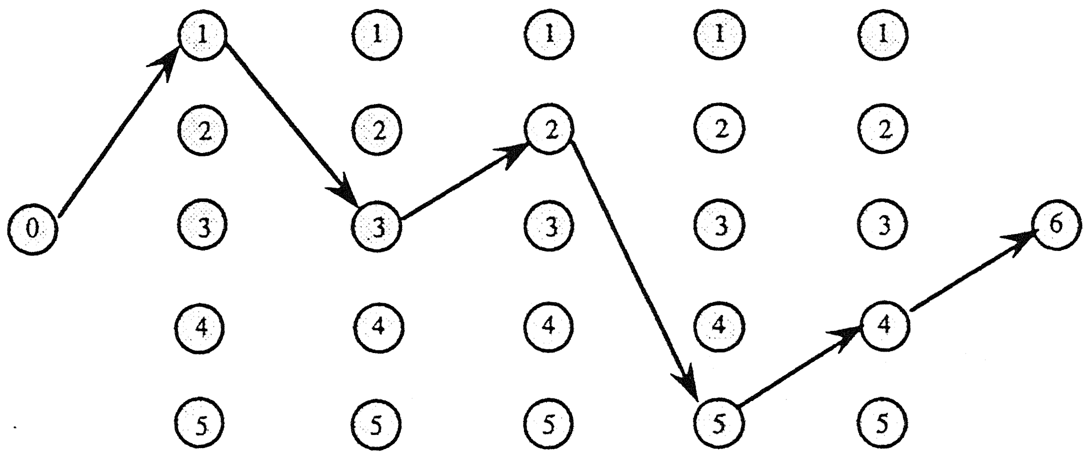


Fig. 3.8. An example tour for Travelling Salesman Problem.

An example tour could be as shown in Figure 3.8. Let x_{ijt} $(i \neq j)$ denote the flow variable associated with arc (i, j) from level 't' . The formulation is as :

$$V = \min \sum_{j=1}^{N} C_{0j}\, x_{0j0} + \sum_{t=1} \sum_{i=1} \sum_{j=1,\, j \neq i} C_{ij}\, x_{ijt} + \sum_{i=1}^{N} C_{i,N+1}\, x_{i,N+1,N}$$

subject to

$$\Sigma_{j=1}^{N} x_{0j0} = 1 \tag{3.8a}$$

$$-x_{0i0} + \Sigma_{j=1,\, j\neq i}^{N} x_{ij1} = 0 \qquad \forall\ i = 1 .. N \tag{3.8ba}$$

$$-\Sigma_{k=1,\, k\neq i}^{N} x_{ki,t-1} + \Sigma_{j=1,\, j\neq i}^{N} x_{ijt} = 0 \qquad \forall\ i = 1 .. N,\ t = 2...N-1 \tag{3.8bb}$$

$$-\Sigma_{k=1,\, k\neq i}^{N} x_{ki,N-1} + x_{i,N+1,N} = 0 \qquad \forall\ i = 1 .. N \tag{3.8bc}$$

$$\Sigma_{i=1}^{N} x_{i,N+1,N} = 1 \tag{3.8c}$$

$$x_{0j0} + \Sigma_{t=1}^{N-1} \Sigma_{i=1,\, i\neq j}^{N} x_{ijt} = 1 \qquad \forall\ j = 1 .. N \tag{3.8d}$$

$$x_{ijt} = 0 \text{ or } 1 \tag{3.8e}$$

Constraints (3.8a)-(3.8c) describe the shortest path problem. Constraints (3.8d) require that each node be visited exactly once and these are the constraints that make the problem combinatorially complex. These constraints are dualized in a Lagrangian fashion. Let π_i be the dual variable associated with jth row of (3.8d). Then the relaxed problem $L^0(\pi)$, given as :

$$L^0(\pi) = \min [\Sigma_{j=1}^{N} \pi_j + \Sigma_{j=1}^{N} (c_{0j}-\pi_j)x_{0j0} + \Sigma_{t=1}^{N-1} \Sigma_{i=1}^{N} \Sigma_{j=1,\, j\neq \iota}^{N} (c_{ij}-\pi_j) + x_{ijt}$$
$$+ \Sigma_{i=1}^{N} c_{i,N+1}\ x_{i,N+1,N}]$$

subject to

(3.8a), (3.8b), (3.8c) and (3.8e)

The relaxed problem is a shortest path problem with arc lengths c_{ij} adjusted to $c_{ij} - \pi_j$, to reflect the relative economy or diseconomy of visiting node j.

Application 3.9 : Scheduling of Power-Generation Systems

Reference : Muckstad and Koenig (1977)

Two major decisions must be made when scheduling the operation of a fossil-fuel power generation system over a short time horizon. First, the "unit commitment" decision indicates what generating units are to be in use at each point in time over the scheduling horizon. This decision must take into account the system capacity considerations and the economic implications of starting up or

shutting down various steam turbines. The "economic dispatch" decision is the allocation of the demand for power or system load among the generating units in operation at any point in time. The model is as follows. Let $x_{it} = 1$ if generating unit 'i' is operating in period 't' and 0 otherwise, for all i = 1...I and t = 1...T, where I is the total number of generating units and T is the total number of periods.

y_{ikt} is the proportion of the maximum available capacity μ_{ik} that is actually used throughout period t, while $k = 1,...,k_i$ and k_i is the number of linear segments in the total production cost curve for unit i. It is obvious that,

$$y_{ikt} > 0 \text{ only if } x_{it} = 1,$$

m_i and μ_i are minimum and maximum operating capacities of generator i, respectively. μ_{ik} is the maximum amount of power that generator i can produce at the kth incremental production cost, g_{ik}. Then the total energy output from generator i in period t can be written as

$$m_i\, x_{it} + \Sigma_{k=1}^{k_i} \mu_{ik}\, y_{ikt}$$

Let $\omega_{it} = 1$ if unit i is started in period t and 0 otherwise. Similarly, $Z_{it} = 1$ if unit i is shutdown in period t and 0 otherwise.

c_i startup cost for generator i

d_i shutdown cost for generator i

g_i cost of operating generator i at its minimum capacity for 1 hour

h_t number of hours in period t

D_t demand level in period t

R_t minimum quantity to be met by planned operating capacity

Then, the model is as :

[P] $\min \Sigma_{i=1}^{I} \{\Sigma_{t=1}^{T} \{c_i\, \omega_{it} + d_i\, z_{it} + h_t\, g_i\, x_{it} + \Sigma_{k=1}^{k_i} \mu_{ik}\, h_t\, g_{ik}\, y_{ikt}\}\}$

subject to

$$\Sigma_{i=1}^{I} (m_i\, x_{it} + \Sigma_{k=1}^{k_i} \mu_{ik}\, y_{ikt}) \geq D_t \quad \text{for } t = 1...T \tag{3.9a}$$

$$\sum_{i=1}^{I} M_i\, x_{it} \geq R_t \quad \text{for} \quad t = 1,..,T \tag{3.9b}$$

$$0 \leq y_{ikt} \leq x_{it} \qquad \forall\ i,t \qquad (3.9c)$$

$$\omega_{it} \geq x_{it} - x_{i,t-1} \qquad \forall\ i, t \qquad (3.9d)$$

$$z_{it} \geq x_{i,t-1} - x_{it} \qquad \forall\ i, t \qquad (3.9e)$$

$$\omega_{it}, z_{it} \geq 0 \qquad (3.9f)$$

$$0 \leq x_{it} \leq 1 \qquad (3.9g)$$

$$x_{it} \text{ integer} \qquad (3.9h)$$

Capacity constraints (3.9a) and reserve constraints (3.9b) are the one that link the generators and are therefore relaxed in Lagrangian fashion. The relaxed problem $R_{u,v}$ decomposes into I single-generator subproblems of the form

$$[R_{u,v}] \qquad \min \Sigma_{t=1}^{I} \{c_i\, \omega_{it} + d_i\, z_{it} + (h_t\, g_i - u_t\, m_i - v_t\, m_i)x_{it} + \Sigma_{k=1}^{k_i} m_{ik}\, (h_t\, g_{ik} - v_t\,)\, y_{ikt}\}$$

subject to

(3.9c) - (3.9h)

Each of these subproblems may be solved by a simple dynamic programming recursion. However, the special structure of the subproblems permits its visualisation as a graph whose nodes represent the operating states of the generator in each period. Figure 3.9 shows a graph that could be used for such solutions.

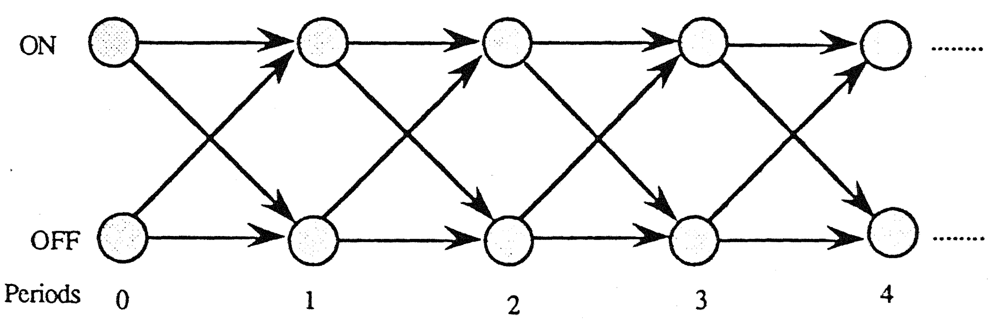


Fig. 3.9. Basic model state graph

As shown, the upper state in each period, represents the 'ON' state and the lower the 'OFF' state. The transition arcs on the graph represent feasible decisions, and the costs associated with these decisions are the arc lengths. A path through this graph specifies an operating schedule for the generator and the problem of finding

the minimum-cost schedule becomes a shortest-path problem on the acyclic state graph.

Application 3.10 : Multi-item Capacitated Lot Sizing Problem

Reference : Chen and Thizy (1990)

This problem consists of determining the quantity and the timing of the production for several products in a finite number of periods (T), so as to satisfy a known demand in each period and minimize the sum of the set-up, production and inventory costs without incurring backlogs. A production capacity is imposed in each period. The costs may vary for each product and for each period. The model is based on a well-known graphic representation of the problem that assigns a node per beginning of period (including T+1), and an arc between each pair of nodes (t, τ+1) representing the possible decision to produce in period t the quantity necessary to satisfy the demands of periods t through τ. For I products, there will be I such graphs. What links them are T capacity constraints. Let x be the decision variable corresponding to an arc between nodes t and τ+1 for product i, which is also the fraction of the demands for periods t through τ of product i that is produced in period t.

$$x_{it} = \Sigma_{\tau=t}^{T} \; x_{it\tau} \, d_{it\tau} \qquad \forall\ i = I...I,\ t = 1...T$$

$$z_{it} = z_i^o + \Sigma_{\tau=t}^{T} x_{it} - d_{i1t} \qquad \forall\ i = I...I,\ t = 1...T$$

where x_{it} and z_{it} are production and inventory carried of product i in period t, respectively.

z_i^o initial inventory

$d_{it\tau} = \Sigma_{j=t}^{\tau} \; d'_{ij}$, where d'_{ij} is the demand for product i in period j adjusted for initial and final inventories.

Let

s_{it} = production set-up cost for production i in period t

$f_{it\tau}$ = total cost of producing product in period t to satisfy demands for periods t through τ, excluding set up costs

c_t = maximum capacity available in period t

$$a_{it\tau} = a_i \, d_{it\tau}$$

$$y_{it} = \{ 1 \text{ if } x_{it} > 0 \,;\, 0 \text{ otherwise} \}$$

where a_i = capacity consumed by the production of one unit of product i.

Then, the shortest path formulation is as :

$$[P] \qquad \min \Sigma_{i=1}^{I} \Sigma_{\tau=1}^{T} \Sigma_{t=\tau}^{T} f_{it\tau} \, x_{it\tau} + \Sigma_{i=1}^{I} \Sigma_{t=1}^{T} s_{it} \, y_{it}$$

subject to

$$\Sigma_{i=1}^{I} \Sigma_{t=1}^{T} a_{it\tau} \, x_{it\tau} \leq c_t \qquad \forall t = 1...T \qquad (3.10a)$$

$$\Sigma_{\tau=1}^{t} x_{i1\tau} = 1 \qquad \forall i = 1...I \qquad (3.10b)$$

$$-\Sigma_{\tau=1}^{t-1} x_{i\tau t-1} + \Sigma_{\tau=t}^{T} x_{it\tau} = 0 \qquad \forall i = 1...I,\ t = 2..T \qquad (3.10c)$$

$$\Sigma_{\substack{\tau=t \\ \tau :\ a_{it\tau}>0}}^{T} x_{it\tau} \leq y_{it} \qquad \forall i = 1...I,\ t = 1..T \qquad (3.10d)$$

$$x_{it\tau} > 0 \qquad \forall\ i=1..T,\ t = 1..T,\ \tau = 1..T \qquad (3.10e)$$

$$y_{it} = 0 \text{ or } 1 \qquad \forall\ i=1..T,\ t = 1..T \qquad (3.10f)$$

Here constraints (3.10a) are capacity constraints. Constraints (3.10b) and (3.10c) are the shortest path equations and (3.10d) are fixed-charge constraints. If we relax capacity and fixed-charge constraints, the problem is decomposed into two subproblems, in which the problem in $x_{it\tau}$ variables is a shortest path problem.

Application 3.11 : Allocation of Inspection Effort on a Production Line

A production line consists of an ordered sequence of production stages and each stage has a manufacturing operation followed by a potential inspection. The product enters stage 1 of the production line in batches of size $B \geq 1$. As the items within a batch move through manufacturing stage, the operations might introduce defects. The probability of introducing a defect at stage i is d_i. It is assumed that all defects are nonreparable and, the defective items are scrapped. After each stage, either all or one of the items can be inspected; it is assumed that the inspection identifies every defective item. The production line must end with an inspection station so that none of the defective items is shipped to the customer. A measure of

taken to inspect the products in the batch fully upto that stage. It should be apparent that with every stage downstream in production, this time tends to be greater. The decision problem is to find an optimal inspection plan that specifies at which stages the items should be inspected so that the total cost of production and inspection is minimized and inspection time is below a certain limit, T. Using fewer inspection stations might decrease inspection costs, but will increase production costs since some unnecessary work might be performed on some units that are already defective. Now, suppose the following cost data is available : 1) p_i, the manufacturing cost per unit in stage i; 2) f_{ij}, the fixed cost of inspecting a batch after stage j, given that the batch was last inspected after stage i and 3) g_{ij}, the variable cost per unit for inspecting one item, given that the batch was last inspected after stage i.

This inspection problem can be formulated as a shortest path problem with an additional constraint on inspection time. The network G(N, A) is defined with (n+1) nodes, numbered 0...n. The network contains an arc (i, j) for each node pair i and j if i < j. Figure 3.11 shows such a network with 4 stages.

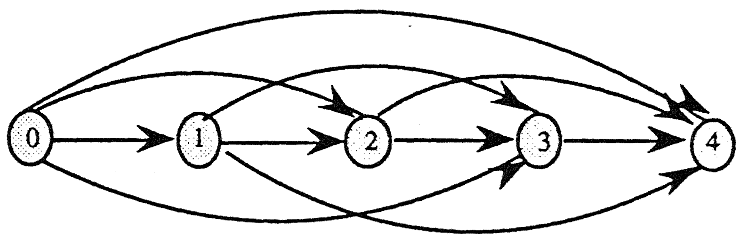


Figure 3.11. Shortest path network associated with the inspection problem

Each path in the network from node 0 to node 4 defines an inspection plan. For example, path 0-2-4 implies that we inspect the batches after stages 2 and 4. Let $B(i) = B\, \pi^{i}_{k=1}\, (1-\alpha_k)$ denote the expected number of nondefective units at the end of stage i. Then, the following cost c_{ij} could be associated with any arc (i, j) in the network :

$$c_{ij} = f_{ij} + B(i)\, g_{ij} + B(i)\, \Sigma^{j}_{k=i+1}\, p_k \qquad (3.11a)$$

Also, let x_{ij} be a binary variable denoting whether or not we select arc (i, j) in the shortest path in the network defined in the above manner and let t_{ij} be the time it takes to inspect a batch after stage j, given that last time batch was inspected after stage i. Then the formulation could be given as

$$[P] \qquad \text{Min} \sum_{(i,j)\in A} c_{ij}\, x_{ij}$$

subject to

$$\sum_{\{j:(i,j)\in A\}} x_{ij} - \sum_{\{j:(i,j)\in A\}} x_{ji} = \begin{cases} 1 & \text{if } i=0 \\ -1 & \text{if } i=n \\ 0 & \text{otherwise} \end{cases} \tag{3.11b}$$

$$\sum_{(i,j)\in A} t_{ij}\, x_{ij} \le T \tag{3.11c}$$

$$x_{ij} \in \{0, 1\} \tag{3.11d}$$

where c_{ij} is as defined in (3.11a).

The Lagrangian relaxation of [P] can be obtained by relaxing constraint (3.11c) with an associated multiplier λ. Then, relaxed problem $[P_\lambda]$ is as

$$[P_\lambda] \qquad \text{Min} \sum_{(i,j)\in A} c_{ij}\, x_{ij} + \lambda\Big(\sum_{(i,j)\in A} t_{ij}\, x_{ij} - T\Big)$$

subject to

(3.11b) and (3.11d)

As shown the relaxed problem is a shortest path problem with cost modified as ($c_{ij} + \lambda t_{ij}$).

CHAPTER 4

APPLICATIONS OF MAXIMUM FLOW PROBLEM

In this chapter, we report the applications in which the relaxed Lagrangian problems are maximum flow (or, equivalently, minimum-cut) problem. We describe the following applications in this chapter :

1. Operations sequencing in discrete parts manufacturing (Bard and Feo [1989])
2. Capital budgeting with limits on annual budget (Mamer and Shogan [1987])
3. Capital budgeting scenario with threshold constraints (Mamer and Shogan [1987])
4. Repair kit selection problem with limits on annual inventory budget (Mamer and Shogan [1987])
5. Repair kit selection subject to a service level constraint (Mamer and Shogan [1987])
6. Selecting freight handling terminals
7. Forest scheduling problem

Application 4.1: Operations Sequencing in Discrete Parts Manufacturing

Reference : Bard and Feo (1989)

It is the problem of sequencing the cutting operation associated with the manufacture of discrete parts on a CNC machine. Two interrelated questions must be answered before prduction can begin in such scenario : (i) in what order should the metal be removed from the blank? (ii) which tools should be used in cutting operations?

The problem is to minimize the production time, and the formulation is as follows :

[P] $\quad F^* = \min F(x, y) = \Sigma_{j=1}^{n} c_j x_j + \Sigma_{t=1}^{r} f_t y_t$

subject to

$$\Sigma_j a_{ij} x_j \geq 1 \qquad \text{for } i = 1 \ldots m \qquad (4.1a)$$

$$\Sigma_{j\in J_t} x_j \le |J_t| y_t \qquad \forall\, t \tag{4.1b}$$

$$\Sigma_{t=1}^{r} y_t \le N \tag{4.1c}$$

$$x_j, y_t = (0, 1) \qquad \forall\, j, t \tag{4.1d}$$

where

$$x_j = \begin{cases} 1 & \text{if path } j \text{ is chosen} \\ 0 & \text{otherwise} \end{cases}$$

$$y_t = \begin{cases} 1 & \text{if tool } t \text{ is chosen} \\ 0 & \text{otherwise} \end{cases}$$

m = number of elemental volumes

n = number of paths

r = number of tools

$$a_{ij} = \begin{cases} 1 & \text{if volume } i \text{ is in path } j \\ 0 & \text{otherwise} \end{cases}$$

N = tool magazine capacity

J_t = set of paths that require tool 't'

c_j = total time it takes to cut path j

f_t = fixed setup and alignment time for tool t.

Constraint (4.1a) ensures that each volume v_i will be removed. Constraint (4.1b) ensures that tool t is in the tool magnazine if it is used for any of the selected paths. Constraint (4.1c) imposes the limit on number of tools used. The Lagrangian relaxation proposed by relaxing (4.1a) and (4.1c) and associating non-negative multipliers λ and μ respectively with each, is as :

$$[P_R] \quad \theta(\lambda, \mu) = \min \Sigma_{j=1}^{n} c_j x_j + \Sigma_{t=1}^{r} f_t y_t + \Sigma_{i=1}^{m} \lambda_i (1- \Sigma_j a_{ij} x_j) + \mu (\Sigma_t y_t - N)$$

$$= \min [\Sigma_{j=1}^{n} (c_j - \Sigma_i \lambda_i a_{ij})x_j + \Sigma_{t=1}^{r} (f_t + \mu) y_t + \Sigma_{i=1}^{m} \lambda_i - \mu N]$$

subject to

$$\sum_{j\in J_t} x_j \le | J_t | y_t \qquad \forall\, t \tag{4.1e}$$

$$x_j, y_t = 0, 1 \qquad \forall\, j, t \qquad (4.1f)$$

λ_i could be interpreted as marginal benefit received from removing volume v_i, $i = 1,..,m$ and μ as the marginal cost of one unit of magazine capacity.

The authors show that the relaxed problem has its primary advantage in its equivalence with another problem which can easily be recast as a minimum cut problem on a bipartite network. The provisioning problem, as it is popularly known, considers a set s_o containing r items from which to choose, each costing $c_t >$ 0 dollars. In this case, the r items are tools with associated costs $c_t = f_t + \mu$. It considers another set of n items from which to choose, each of which confers special benefits $B_j \geq 0$. In this case, these n items are paths and for each path, $B_j = \Sigma_i \lambda_i a_{ij} - c_{ij}$. A better understanding of the representation of P_R in the form of the desired bipartite network can be had by referring to Figure 4.1.

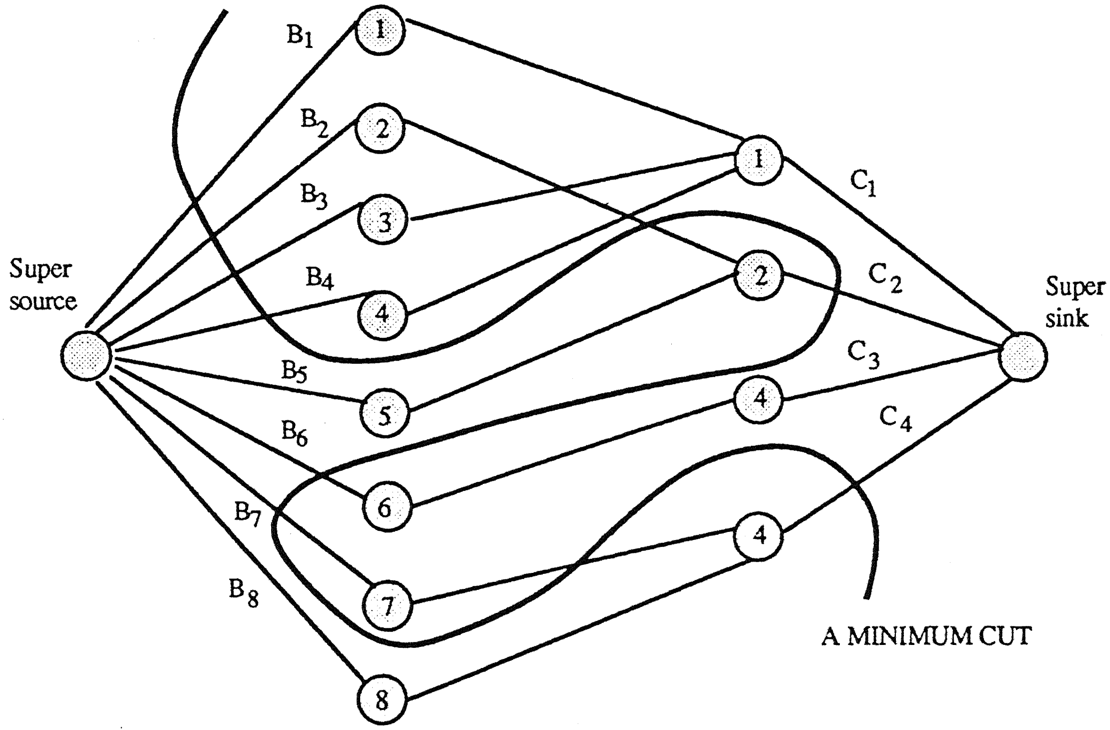


Fig. 4.1. Operations sequencing as Maximum Flow Problem.

As shown in the figure, there are nodes for paths and for tools. The former are connected to a supersource by arcs with capacity B_j and the latter are connected to supersink by arcs with capacity c_t. The arcs (with infinite flow capacity) between the two sets of nodes arise in the case when a tool is used by a particular path. For example, tool 2 is used by paths 2 and 5. A possible minimum cut could be as shown in Figure 4.1.

Application 4.2 : Capital Budgeting with Limits on Annual Budget

Reference : Mamer and Shogan (1987)

The capital budgeting problem is considered with following characteristics :

1. There are M potential project, each of which must be taken fully or not.

2. The undertaking of project i requires the performance of a known subset S_i of a set of N activities. Set S_i is referred to as project i's enabling set. The different enabling sets may not be mutually exclusive.

3. c_j is the cost of performing activity j.

4. b_i is the revenue earned by undertaking project i.

5. The performance of activity j consumes a_j units of resource whose total consumption cannot exceed A units.

Let

$$x_j = \begin{cases} 1 & \text{if activity j is performed} \\ 0 & \text{otherwise} \end{cases}$$

$$y_i = \begin{cases} 1 & \text{if project i is undertaken} \\ 0 & \text{otherwise} \end{cases}$$

The capital budgeting problem is to select projects which maximize profit subject to the resource limitation. The formulation is as :

$$\text{Maximize } \Sigma_{i=1}^{M} b_i y_i - \Sigma_{j=1}^{N} c_j x_j$$

subject to

$$\Sigma_{j=1}^{N} a_j x_j \leq A \qquad (4.2a)$$

$$y_i \leq x_j \qquad \forall j \in S_i \text{ and } 1 \leq i \leq M \qquad (4.2b)$$

The problem when reformulated in equivalent minimization form becomes

$$\text{Minimize } \Sigma_{j=1}^{N} c_j x_j + \Sigma_{i=1}^{M} b_i (1 - y_i)$$

subject to

$$\Sigma_{j=1}^{N} a_j x_j \leq A \qquad (4.2c)$$

$$y_i \leq x_j \qquad \forall j \in S_i \text{ and } 1 \leq i \leq m \qquad (4.2d)$$

The first term of the objective function represents the total fixed costs of the activities performed, and the second term represents the total opportunity costs for the projects not undertaken.

Without the resource constraint (4.2c), the problem simplifies to one referred to as Problem (NET). Problem (NET) was first considered by Rhys (1970) and solved elegantly by Balinski (1970), who showed its equivalence to maximum flow problem on the network as shown in Figure 4.2. Thus, the problem is of the form of maximum flow problem with the additional constraint on resource, which in this case is the annual budget available for carrying inventory. This additional constraint is relaxed in a Lagrangian fashion by associating non-negative multiplier λ with it and bringing into the objective function. This relaxed problem is solved as a maximum flow problem.

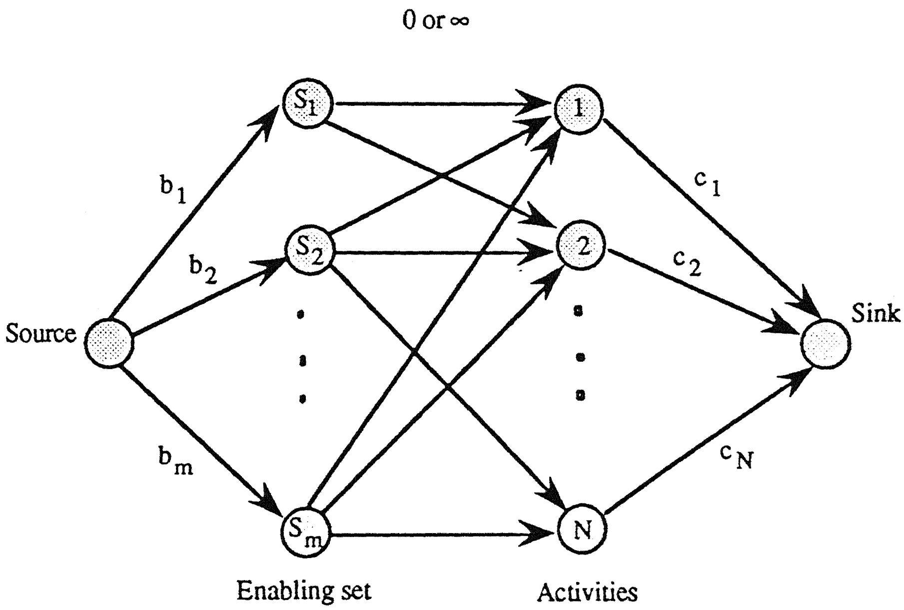


Fig. 4.2. Flow network of problem (NET)

As shown in the Figure 4.2, the arcs from enabling set nodes to activites nodes have a capacity of ∞ if the activity is in that enabling set and is zero otherwise.

Application 4.3 : Capital Budgeting Scenario with Threshold Constraints

Reference : Mamer and Shogan (1987)

This is a slight variant of problem of capital budgeting with limits on annual budget. It is same as Application 4.2 with the resource constraint $\Sigma_{j=1}^{N} a_j x_j \leq A$ replaced by the "threshold constraints" $\Sigma_{i=1}^{M} a_i y_i \geq A$. In the capital budgeting scenario, this means A now represents the minimum permissible "level of service"

to be provided by the projects undertaken, and a_i represents project i's contributi to the service level threshold. Once this additional constraint has been considere the rest of the problem can again be shown as being equivalent to the Problem [NE and consequently to a maximum flow network. The "level of service" constrain are relaxed in a Lagrangian fashion and one can proceed in a similar manner as th of Application 4.2.

Application 4.4 : Repair Kit Selection Problem with Limits on Annual Inventor Budget

Reference : Mamer and Shogan (1987)

The repair kit selection model was originally proposed by Mamer and Smitl (1982). Their work involved the following scenario :

A repair facility must decide which parts (and tools) to include in the standarc field repair kit. Each repairman will use the standard kit for a single service call anc will return to the repair facility to have the kit restocked before proceeding to the next service call. If the service call requires parts not included in the kit, the repairman will either return to the repair facility to pick up the required parts or bring the failed item to the facility for repair. Assuming N parts are available for inclusion, stocking the repair kit is equivalent to choosing a subset k of the set of N parts. Inclusion of part j in the repair kit results in an annual inventory cost of c_j. When using the kit on a service call, the repairman will encounter a breakdown of any one of M types. Breakdown i occurs with an annual frequency of f_i times per year and, at each occurence, requires a subset S_i of the set of N parts. Thus, on-site repair of a breakdown of type i is possible if and only if $S_i \subseteq k$. If the repair kit lacks at least one part required to repair a breakdown of type i, then a penalty cost of p_i is incurred whenever the repairman encounters such a breakdown. The objective of the service facility is to stock the standard repair kit in such a way as to minimize the sum of total expected annual penalty costs and the total annual inventory costs. There is an additional constraint on annual inventory budget.

Let

$$x_j = \begin{cases} 1 & \text{if part j is included in the kit} \\ 0 & \text{otherwise} \end{cases}$$

$$y_i = \begin{cases} 1 & \text{if kit contains every part required for} \\ & \text{an on-sit repair of breakdown type i} \\ 0 & \text{otherwise} \end{cases}$$

Then the formulation proposed is as :

$$\text{Minimize } \Sigma_{i=1}^{M} f_i (1 - y_i)$$

subject to

$$\Sigma_{j=1}^{N} c_j x_j \leq C \tag{4.4a}$$

$$y_i \leq x_j \qquad \forall j \in S_i \text{ and } 1 \leq i \leq M \tag{4.4b}$$

The objective is to minimize annual frequency of breakdown where C is the total annual inventory budget. The authors show the equivalence of this problem with that of Application 4.2 after constraint (4.4a) is relaxed in a Lagrangian fashion. Hence, in this case also, the relaxed problem can be solved as a maximum flow problem subject to a side constraint.

Application 4.5 : Repair Kit Selection Subject to a Service Level Constraint

Reference : Mamer and Shogan (1987)

This is a slight variant of the repair kit selection model of Application 4.4 in the way the objective function and the additional constraint are proposed. Here, the objective is to minimize the total annual inventory costs, subject to a service level constraint which specifies that least F percent breakdowns should be fixable with the repair kit. Therefore, following the same notations as that of Application 4.4, the formulation could be represented as :

$$\text{Minimize } \Sigma_{j=1}^{N} c_j x_j$$

subject to

$$\Sigma_{i=1}^{M} f_i y_i \geq F (\Sigma_{i=1}^{M} f_i) \tag{4.5a}$$

$$y_i \leq x_j \qquad \forall j \in S_i \text{ and } 1 \leq i \leq M \tag{4.5b}$$

One can proceed in the similar fashion as that of Application 4.4 for this problem as well.

Application 4.6 : Selecting Freight Handling Terminals

A transport company is considering the installation of a number of freight freight handling terminals. It wants to choose from a set S of possible locations for the terminals. The firm has the potential to attract market share (which is a given

amount of demand) between some of the pairs of terminal to satisfy the demand between locations i and j, the company must locate terminals, at both of these locations. Let c_j be the cost of installing a terminal at location j and p_{ij} be the profit obtained by satisfying the demand between location i and j. Consider, for example, the network shown in Figure 4.6a.

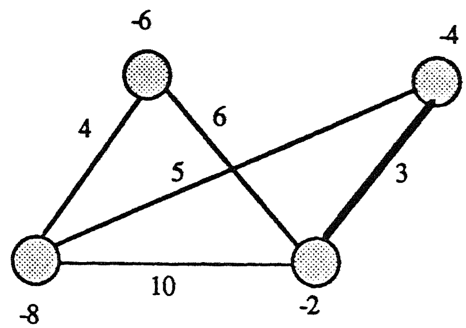


Fig. 4.6a Selection Problem

Each node in this network represents a potential terminal, the number next to it represents the negative of the cost of installing that terminal. Each arc (i, j) represents a service that can be operated only if both the terminal i and j are operating; the number next to an arc represent the profit obtained by operating that service. Suppose, we decide to operate terminals 1, 2 and 3; then we can operate service only between the following pairs of terminals : {(1, 2), (1, 3) and (2, 3)}.

Next, suppose P be the minimum amount of profit that the company is interested in making while being in the business, i.e., the profit earned by installing terminals at i and j stationed over all such (i, j) pairs should exceed P. This type of "threshhold constraint" destroys a special structure of the problem, which can very easily be transformed to an equivalent network. Hence, if we relax the threshhold constraint in a Lagrangian fashion, we can solve the relaxed network flow problem to generate effective lower bounds to the original problem. We now show an equivalent network formulation of the relaxed problem. Let us define a bipartite network $G = (N_1 \cup N_2, A)$ with a node in N_1 for every service and a node in N_2 for every terminal. The service node representing the service between nodes i and j has two outgoing arcs entering the nodes, representing the terminals i and j, implying that whenever we decide to provide the service between i and j and accrue the profit p_{ij}, we must install the terminal at these two nodes and incur the installation cost C_i and C_j. Figure 4.6b shows the resulting maximum weight closed problem {for the network of Figure 4.6a}.

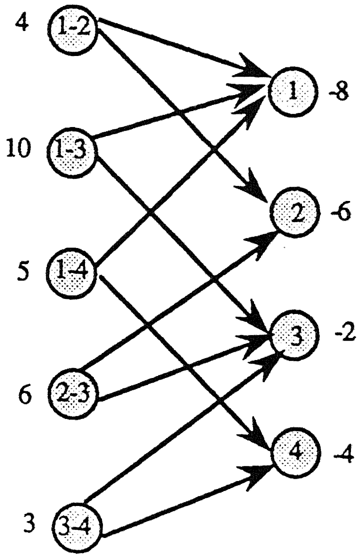


Fig. 4.6b. The corresponding Maximum Closure Problem.

This network can easily be transformed to an equivalent maximum flow problem, if we add a source s and sink t to the network and add the arcs (s, i) for all i with weight $w_i > 0$, and arcs (j, t) for all j with weight $w_j < 0$. The capacity of the arcs (s, i) is w_i and and arcs (j, t) is $-w_j$ and capacity of intermediate arcs is infinity. Hence, in this case the relaxed problem is a maximum flow problem.

Application 4.7 : Forest Scheduling Problem

Paper and wood product companies need to define cutting schedules that will maximize the total yield of their forests over some planning period. A particular company with control of β units wants to identify the best cutting schedule over a planning horizon of k years. Forest unit i has a total acerage of a_i units. Studies that the company has undertaken predict that unit i will have w_{ij} tons of wood available in the jth year. Based on its prediction of economic conditions, the company believes that it should harvest at least l_j tons of wood in year j (from all the forest units). Due to the availability of equipments and personnel, the company can harvest at most u_j tons of wood in year j. The company incurs a cost c_{ij} per unit acerage to harvest wood in forest unit i in year j. The company has a fixed budget C at the starting of the planning period for the harvesting purposes and the expenditure on harvesting in all the forest units for all years should not exceed this limit. In what follows, a simple network model is associated with the problem above. As shown in Figure 4.7, two sets of node are considered, one each for forest units (i) and respective years (j), in planning period. Source (s) and sink (t) is then added to the network and arcs are added from node s to nodes i and from nodes j to t.

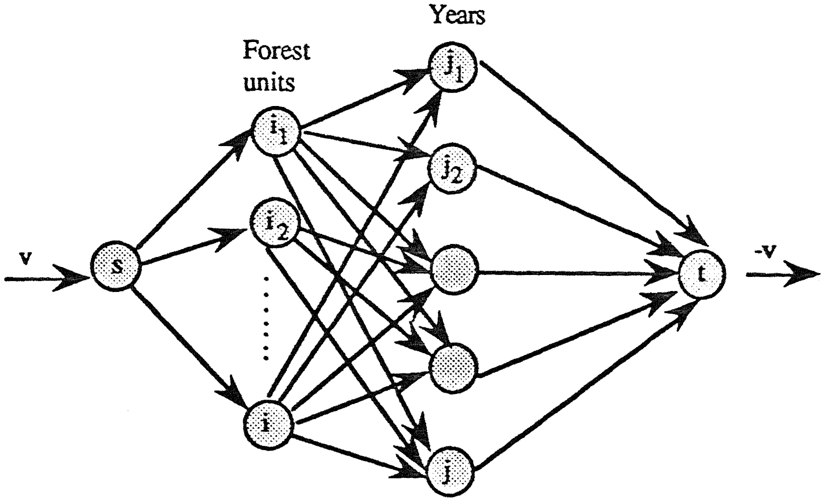


Fig. 4.7. Forest scheduling as Maximum Flow Problem.

The arcs (s, i) have zero cost and capacity equal to a_i and arcs (j, t) have zero cost and, lower and upper bounds on flow as l_j and u_j units, respectively. The arcs (i, j) have an associated cost c_{ij} and capacity equal to w_{ij}. Note that for each forest unit, there is an outgoing arc for all the years.

Having defined the network in the above way, it is easy to see that maximizing the yield over all the years is equivalent to finding a maximum flow in the above network subject to the constraint that expenditure over all the years is less than C. If x_{ij} is the flow on the arc (i, j) of the above network, then this constraint could be written as

$$\Sigma_{(i,\, j)\in A}\; c_{ij}\, x_{ij} \leq C \tag{4.7a}$$

where A is the arc set of this graph. This side constraint makes the problem hard by destroying the network structure embedded in the constraint matrix. Dualizing this constraint in a Lagrangian fashion by associating non-negative Lagrangian multipliers with it results in the relaxed problem that is easily solvable by the efficient algorithms available for maximum flow problem.

CHAPTER 5

APPLICATIONS OF MINIMUM COST FLOW PROBLEM

In this chapter, we report applications in which the Lagrangian subproblems are minimum cost flow problem. We describe the following applications in this chapter :

1. Multilocation facility modernisation problem
   (Mason, Girard and Gu [1990])
2. Manpower planning problem with attrition constraint
   (Price and Gravel [1984])
3. Manpower planning problem with budget constraint
   (Price and Gravel [1984])
4. Routing of container ships
   (Rana and Vickson [1991])
5. Network flow problems with variable upper bounds
   (Shetty [1990])
6. Multicommodity flow problem
   (Saviozzi [1986])
7. Set partitioning problem
   (Ali and Thiagarajan [1987])
8. The equal flow problem
   (Ali, Kennington and Shetty [1987])
9. Node-weighted Steiner tree problem
   (Segev [1987])

Application 5.1 : Multilocation Facility Modernisation Problem

Reference : Mason, Girard and Gu (1990)

Telecommunications and other public utilities are frequently faced with the task of modernising their facilities as a result of technological advance and/or demand for new services. The owner of these facilities seeks a modernisation policy which optimizes an economic indicator (most common being NPV), while satisfying all budgetary and service restrictions.

The problem in its most general setting finds a typical example in local area telecommunication network, where each new customer has to be connected to an office by a dedicated line. The provisioning is done at fixed intervals, hence the model will have discrete time nature.

New customers can be connected to 3 types of services : P, E or R.

P , i.e., Plain is simplest and uses either analog (A) or digital (D) transmissions.

E , i.e., enhanced is telephone line upgraded for digital rates longer than what would be possible with a plain line.

R , i.e., replaced gives even longer rates and more sophisticated services.

At any instance, a line is characterized by its line type (or kind of transmission) and the service it uses, i.e., by {A, D} x {P, E, R}. Any such pair would be state of the line. Given a known demand at any given time the question faced by the planner is : How to meet the demand? This decision has to be taken for each office in the network, here called a location, and is independent except for the presence of global limits on the capital available for the modernization. These budget constraints are the complicating elements of the multilocation problem. The model description is as follows :

INPUT PARAMETERS

N number of states per location, in this case, these belong to the set {PA, PD, EA, ED, RD}

L the number of locations

T number of time periods

$a^l(t)$ total new demand in period t at location l

$b_i^l(t)$ net cash flow per line introduced at location l in period t into state i

$d_i^l(t)$ net cash flow per line operating in state i between period t and t+1 at location l

$c_{ij}^l(t)$ net cash flow per line for the transition from i to j at beginning of period t in location l

DECISION VARIABLES

$a_i^l(t)$ number of new lines introduced in state i at the beginning of period 't' at location l

$x_{ij}^l(t)$ number of lines undergoing the transition from state i to state j at the beginning of period t at location l

$y_i^l(t)$ number of lines that remain in state i between period t and t+1 at location l.

$y_i^l(0)$ = 0, by convention.

COST FUNCTION

In the objective (or cost) function, NPV is tried to be maximized :-

$$Z^* = \max_{x,y,a} Z = \Sigma_{l=1}^{L}\Sigma_{t=1}^{T} [\sum_{(i,j)\in T} x_{ij}^l(t)\, c_{ij}^l(t) + \Sigma_i (y_i^l(t)\, d_i^l(t) + a_i^l(t)\, b_i^l(t))] \quad \text{... (5.1a)}$$

CONSTRAINTS

If D(t) is total budget available at the beginning of period t, budget constraint is as :

$$- \Sigma_{l=1}^{L} \sum_{(i,j)\in T} x_{ij}^l(t)\, c_{ij}^l(t) + a_i^l(t)\, b_i^l(t) \le D(t) \quad \text{for } t = 1..T \qquad (5.1b)$$

These are lower and upper bounds on number of lines that can undergo transition between two periods, namely $w_{ij}^l(t)$ and $v_{ij}^l(t)$; and also on number of lines that remain in the same state; namely $p_i^l(t)$ and $q_i^l(t)$. Therefore,

$$v_{ij}^l(t) \le x_{ij}^l(t) \le w_{ij}^l(t) \qquad t = 1..T,\ \forall\ (i, j),\ l = 1..L \qquad (5.1c)$$

$$p_i^l(t) \le y_i^l(t) \le q_i^l(t) \qquad t = 1..T,\ i = 1..N, l = 1..L \tag{5.1d}$$

Finally, no new demand should go without service

$$a^l(t) = \Sigma_{i=1}^{N} a_i^l(t) \tag{5.1e}$$

Similarly, lines that are going to be upgraded should not disappear and hence the conservation equation :

$$\sum_{(i,j)\in T} x_{ij}^l(t) + y_i^l(t) - y_i^l(t-1) - a_i^l(t) = 0 \quad t = 1..T, l = 1..L, i = 1..N \tag{5.1f}$$

Finally, we have the following usual conditions.

$$x_{ij}^l(t) \ge 0 \quad \text{integer} \tag{5.1g}$$

$$y_i^l(t) \ge 0 \quad \text{integer} \tag{5.1h}$$

On the time-expanded network, the model is multicommodity flow made up of T transition networks. The complicating constraints here is the budget constraint (5.1b), which is given a price (Lagrangian multiplier u_t associated with t^{th} constraint (5.1b)) and is incorporated into objective function.

The Lagrangian function L(u) can be written as :

$$L(u) = \max_{x,y,a} [Z + \Sigma_{t=1}^{T} u_t (\Sigma_{l=1}^{L} \Sigma_{i,j} x_{ij}^l(t)\, c_{ij}^l(t) + \Sigma_i a_i^l(t)\, b_i^l(t) + D(t)]$$

subject to

constraints (5.1e) - (5.1h)

The task is to obtain smallest upper bound, that is, to solve the problem

$$\min_u L(u) \text{ subject to } u \ge 0 \tag{5.1i}$$

The authors show that the interest of the method lies in the fact that this problem separates into L independent single-location problem (Again, the term 'Location' here refers to an office site), since the budget constraints have now been uncorporated into objective function with suitable multipliers. Because of the form of constraints (5.1c)-(5.1d) each of these subproblems is an ordinary minimum cost flow problem that can be solved by any minimum cost flow algorithm. A more detailed description of the above can be found by referring to the original paper by Mason, Girard and Gu (1990).

Application 5.2 : Manpower Planning Problems with Attrition Constraint

Reference : Price and Gravel (1984)

A certain class of manpower problem is often modelled using goal-programming. The authors refer to the paper by Price (1978), which discribes this problem in the following way :

Consider a hierarchy (see Figure 5.2a) divided vertically by rank and horizontally by occupational classification or trade.

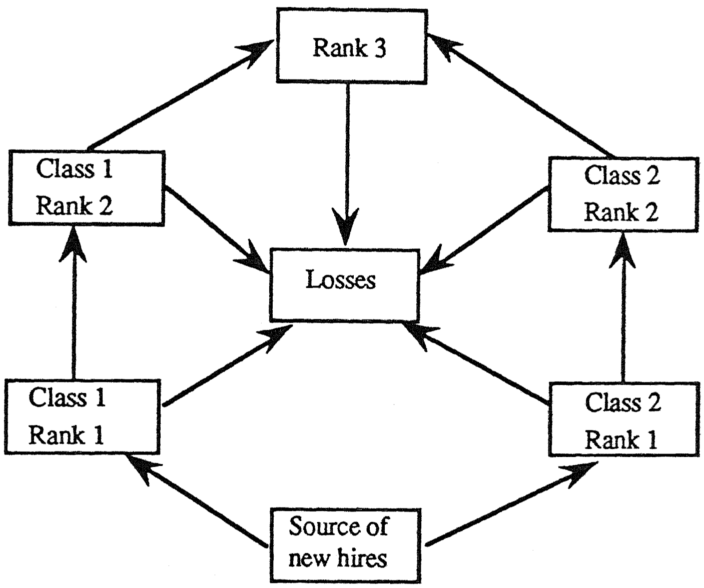


Fig. 5.2a. Three ranks, two classification system.

Hiring is possible only into the lowest rank, and promotions are only to next higher rank. Demotions are neglected, and attritions can be forecast. It is desired to find a hiring and promotions plan for the next time period that honours constraints on :

1. numbers in each rank/classification cell

2. numbers promoted from one level to another

3. total number in a given rank

4. total number promoted from one rank to another.

Let us consider how this problem can be represented as a network flow problem.

$x_{ij}(t)$ strength of rank i, classification j, at the end of period t

$y_{ij}(t)$ the number promoted out of rank i (to rank i+1), classification j during period t

$\phi_{ij}(t)$ lower bound on $x_{ij}(t)$

$\alpha_{ij}(t)$ upper bound on $x_{ij}(t)$

$\tau_{ij}(t)$ lower bound on $y_{ij}(t)$

$\beta_{ij}(t)$ upper bound on $y_{ij}(t)$

$a_{ij}(t)$ attrition from rank i, clasification j, during period t.

Now, the following constraints are expressed.

Manning level in each rank/classification state $\phi_{ij} \leq x_{ij}(t) \leq \alpha_{ij}$

Promotion level from each rank/classification state $\tau_{ij} \leq y_{ij}(t) \leq \beta_{ij}$

Manpower accounting constraints $x_{ij}(t-1) - a_{ij}(t) - y_{ij}(t) + y_{i-1,j}(t) = x_{ij}(t)$

Now, a representative network of the above could be shown as in Figure 5.2b.

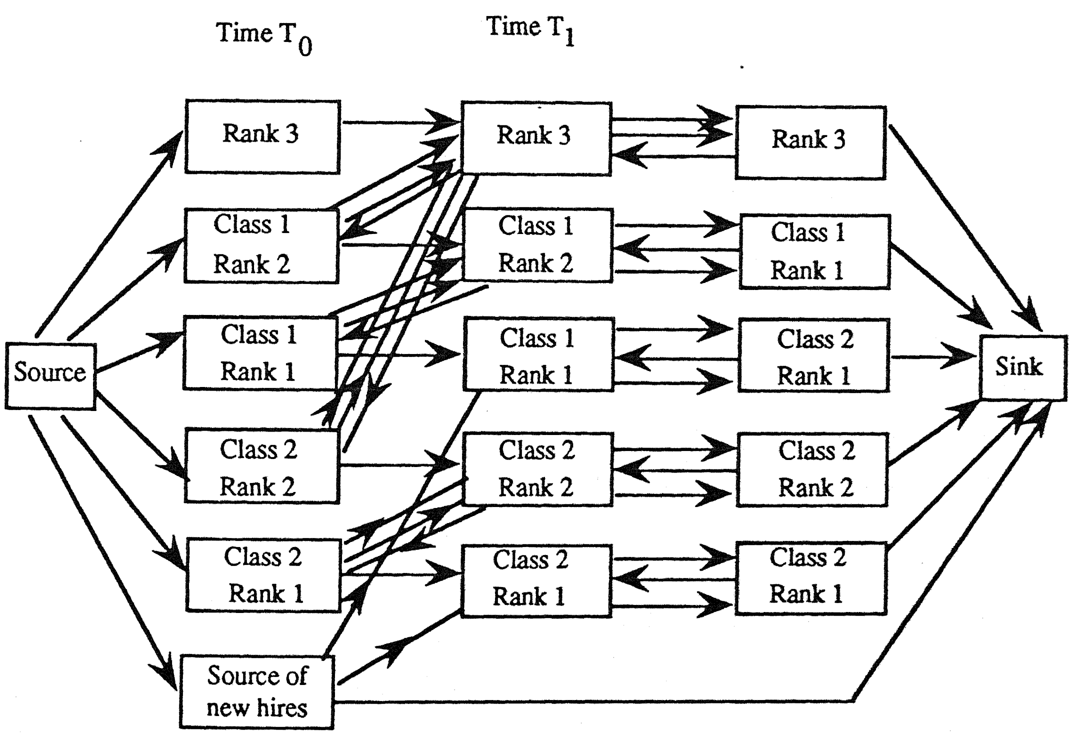
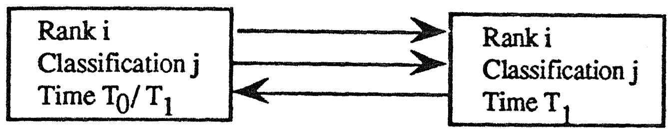


Fig. 5.2b Network Representation. Two classification, three ranks, one time transition.

The triplets of arcs are used to represent the goal formulation. Figure 5.2c illustrates how this is done. The upper arc carries all flow that is between desired flow limits and bears no cost. The second arc, at the cost c_2 bears any overflow that must be carried to meet overall goals. The third arc, at cost c_3 and in opposite direction to the first two, will bear any flow needed to compensate for the lower bound on the first arc.



Arc	Lower bound	Upper bound	Cost
1	ϕ_{ij}	α_{ij}	0
2	0	∞	C_2
3	0	∞	C_3

Fig. 5.2c. Representing the goal formulation

Refering back to Figure 5.2b, note that a supersource and a supersink (shown as network source and sink) are added to complete the network and arcs have been added from various nodes that have capacities as existing state at time T_0 at that node. The arc connecting to source of new hires has infinite capacity.

The attrition constraint. For the network extending over several time-periods, one would like to address to the problem of attrition, i.e., losses due to retirements and voluntary departures. To accomodate this, the attrition flow out of a node is fixed at some percentage (p) of the total flow representing retired manpower. This would introduce ratio constraints of the form

$$a_{ij}(t) = p\, x_{ij}(t)$$

where p is the attrition rate.

The addition of such a constraint destroys the network structure of the problem and we will address to this problem by Lagrangian relaxation.

In general, the complete problem including both network structure and side constraint can be expressed as :

(P) Minimize cx

subject to

$$Ax \leq b \tag{5.2a}$$

$$Sx \leq d \tag{5.2b}$$

$$x \geq 0 \tag{5.2c}$$

Let constraint (5.2a) define the network structure of the problem and matrix S contain the side constraint. Then, the relaxed problem is

(PR_λ) Minimize $(cx + \lambda(d\text{-}Sx))$

subject to

$$Ax \leq b \tag{5.2d}$$

$$x \geq 0, \lambda \geq 0 \tag{5.2e}$$

which can be solved as a minimum cost flow problem with cost vectors being (c-S). Now, the subgradient approach can be used in the usual manner to obtain tight lower bounds on the attrition problem.

Application 5.3 : Manpower Planning Problems With Budget Constraint

Reference : Price and Gravel (1984)

An aspect not covered by the manpower problem with attrition constraint is a means of maintaining direct financial control. One might wish to ensure that in a given time period, some overall budgetary restrictions were honoured. This could be done by adding the constraint

$$\sum_{(i,j \in H)} x_{ij}(t) \,.\, s_{ij}(t) \leq S(t) \tag{5.3a}$$

where S(t) is the budget available in period t, $s_{ij}(t)$ is the salary paid in the category ij in period t, and H is the set of arcs representing manpower levels in period t. Bundle constraint of this kind when added to the manpower planning problem destroy its network structure. In addition to the constraints on salary budgets, other

bundle constraints may also exist. One might wish, for example, to control the total numbers hired, or in some other way restrict the weighted sum of flows in the arcs. Any such constraints may be grouped and treated by the Lagrangian relaxation techniques as presented in the Application 5.2.

Application 5.4 : Routing of Container Ships

Reference : Rana and Vickson (1991)

Seaborne shipping is considered to be the major artery of international trade. In international shipping, a large number of consignments coming from many different sources make a shipload of general cargo. Containers loaded at one port, in general, have more than one port of destination, and may not fill a ship. Since this model considers both pickup and delivery at a port, it differs from the travelling salesman or vehicle routing problem. The problem of scheduling container ships, then, is to find the optimal route for each ship and the cargo it carries between any port pair and to calculate the frequency of service provided for each port. Even without including such stochastic elements as weather conditions, delays or waiting times, the problem is quite large and complex.

A ship travels in one direction, loading and discharging cargo at the intermediate and end ports. It may visit ports on its Eastbound voyage or the Westbound voyage, which depends on the amount of cargo available at respective ports. Availability of cargo is determined at weekly basis and accordingly a schedule is determined and the shippers are informed. The typical network associated with container trade is as shown in the Figure 5.4, and it is obvious that it is different from type of networks used in vehicle routing.

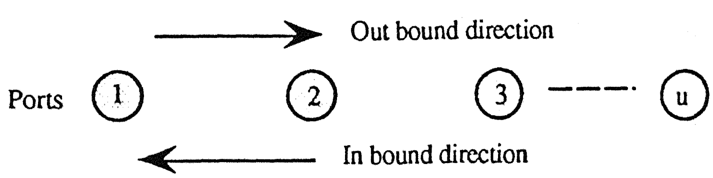


Fig. 5.4. Network of ports

We give the description of the mathematical formulation without giving the exact model, because of the length and the complexity of the model (Interested reader may refer to original paper for more detailed view on the same).

The objective is to maximize the net profit, i.e., revenue minus operating costs, from multiple shifts alloted to a network of ports. The constraints are as follows :

1. Capacity constraints - Ship's capacity cannot be exceeded on any voyage segment.

2. Departure and Arrival constraints - If the ship is not visiting either the port of origin of some cargo or the port of the cargo's destination, the cargo cannot be transported.

3. Cargo constraints - Total cargo transported from port i to port j cannot exceed the available cargo.

4. Network constraints - These constraints serve two main requirements. First, a ship must make a connected trip and second, it must have an option to stop at a port or not.

5. Route time constraints - The total time spent by a ship in making an integer number of trips should not exceed the planning horizon.

The authors show that by relaxing the complicating constraints on capacity, i.e., constraint number (1), in a Lagrangian fashion, the problem decomposes into v subproblems, here v is the total number of ships to be assigned to the network. Each subproblem, called SP_k for $k = 1,..,v$ is treated further by decomposing the underlying network in such a way that a fixed source and destination pair is considered at a time. This subproblem is treated further by a specialized algorithm which decomposes it into two subproblems, namely, CAS (Cargo Allocation Subprograms) and INS (Integer Network Subprograms), where the latter can be solved by any minimum cost network flow algorithm.

Application 5.5 : Network Flow Problems with Variable Upper Bounds

Reference : Shetty (1990)

The network problem with variable upper bounds (abbreviated as NVUB) is easily conceptualized as a minimum cost network flow problem with variable upper bounds on certain arcs. The problem is represented by an nxm node-arc incidence matrix, A, in which k arcs are identified and required to have variable upper bounds. The columns of A are so arranged that the first k columns (A_1) corresponds to VUB arcs and the remaining m-k columns (A_2) to others. Mathematically, NVUB is represented as :

$$\text{Min } cx + d\sigma + ez$$

subject to

$$A_1x + A_1\sigma = b \tag{5.5a}$$

$$0 \le x \le z \le u \tag{5.5b}$$

$$0 \le \sigma \le v \tag{5.5c}$$

$$x, \sigma, z, \text{ integer} \tag{5.5d}$$

Here, [c, d] is a lxm vector of unit costs, b is nxl vector of node requirements, 0 is an approximately dimensioned vector of zeroes, [u, v] is an mxl vector of upper bounds, and [x, σ, z] is the vector of decision variables. Constraints (5.5b), also known as the side constraints, represent variable upper bounds in which each arc in x is bounded by a decision variable in z. NVUB belongs to the class of NP-complete problems, therefore, effective heuristics based on Lagrangian relaxation and decomposition, have been proposed in the following manner.

A lower bound on the objective function of NVUB can be obtained by using the Lagrangian dual of the the problem exploiting the embedded network structure of the problem. Associating the Lagrangian multiplier ω_k with the kth side constraint and defining the k-vector $\omega = (\omega_1..\omega_k)$, the Lagrangian dual for NVUB, referred to as LNVUB, may be stated as

[LNVUB] $\quad \underset{\omega \ge 0}{\text{Max}}\ H(\omega)$

$$\text{where } H(\omega) = \min\{cx + d\sigma + ez + \omega(x\text{-}z)\} \tag{5.5d}$$

subject to

constraints (5.5a), (5.5c) and (5.5d)

This problem separates into subproblem of variables of z and x, respectively. The optimal value of subproblem in z can be found by inspection. However, the optimal values of subproblem in x require solution of a minimum cost network flow problem given by :

$$\min \{(c+\omega)x + d\sigma \mid A_1x + A_2\sigma = b,\ \ 0 \le x \le u,\ \ 0 \le \sigma \le v\}.$$

Thus, LNVUB provides effective lower bound to NVUB.

Application 5.6 Multicommodity Flow Problem

Reference : Saviozzi (1986)

The author proposes the idea of using Lagrangian relaxation to provide advance start for the multicommodity network flow problem. Multicommodity network flow problem arises in a variety of applications, some of which are routing of multiple commodities on a network, multilevel tanker scheduling, warehousing of seasonal products, etc. For a given node-arc incidence matrix A representing a directed graph, the minimization problem P to be solved for multicommodity flow problem can be stated mathematically as

$$[P] \qquad \min \sum_k c^k x^k$$

subject to

$$Ax^k = r^k \qquad \forall\ k=1..\ K \tag{5.6a}$$

$$\sum_k x^k \leq b \tag{5.6b}$$

$$0 \leq x^k \leq u^k \qquad \forall\ k = 1,.., K \tag{5.6c}$$

where x^k, c^k, u^k, r^k are, respectively, the flow, cost, requirement and capacity vectors for commodity k; and b is the arc capacity vector to be shared among the k commodities.

The problem P can be represented for k commodities as a minimum cost network flow problem for each commodity with (5.6a) and (5.6c) as constraints, but for the constraint (5.6b) that is, binding on all the commodities to share mutual arc capacity. The Lagrangian relaxation scheme is, therefore, used effectively in this case, where the independent subproblems are minimum cost network flow problems.

The following maximum problem is proposed :

$$\max_{\omega \geq 0} \{L(\omega)\}$$

$$= \max_{\omega \geq 0} \min_x \{ \sum_k c^k x^k + \omega(\sum_k x^k - b) \mid Ax^k = r^k, 0 \leq x^k \leq u^k\}$$

$$= \max_{\omega \geq 0} \{-\omega b + \sum_k \min_x (c^k + \omega)\ \alpha^k \mid Ax^k = r^k, 0 \leq x^k \leq u^k\}$$

This is obtained by dualizing capacity constraints (5.6b) in a Lagrangian fashion and associating with it a multiplier vector ω. The updated costs are $(c^k + \omega)$. The resulting subproblem is, thus, minimum cost network flow problem.

Applicaton 5.7 : Set Partitioning Problem

Reference : Ali and Thiagarajan (1987)

Set partitioning problems are perhaps the most widely encountered of the class of binary integer programs. The set partitioning problem is stated as $\{\min cx \mid Ex = e, x \text{ binary}\}$, where E is an mxn binary element-set incidence matrix with $e_{ij} = 1$ if element i belongs to set j and 0 otherwise, and e is an m-vector of 1's. The solution of these problems is based on the fact that the hidden network structure in the set partitioning constraint matrix, E, can easily be exploited. If the constraint matrix E does not have any apparent structure, the question of existence of hidden special structure arises. However, E can be exploited to have completely hidden structure in very specific application only. The set partitioning problem (SPP) as explained above can also be written as

[SPP] $\min cx$

subject to

$$E_1 x = e_1 \tag{5.7a}$$

$$E_2 x = e_2 \tag{5.7b}$$

$$x \text{ binary} \tag{5.7c}$$

where $m_1$xn submatrix E_1 is a row submatrix transformable to a network. The $m_2$xn submatrix E_2 forms the remaining constraints. It can, therefore, be shown that SPP is equivalent to a problem [NS], given as :

[NS] $\min cx$

subject to

$$Ax = b \tag{5.7d}$$

$$Sx = e \tag{5.7e}$$

x binary (5.7f)

where A is node-arc incidence and S is the set of side constraints.

The solution strategy using Lagrangian relaxation is as follows. The Lagrangian function, L_λ, is obtained by dualizing non-network side constraints SX = e, where λ is the vector of associated dual variables and given by

$$L_\lambda = \min \{cx + \lambda(Sx - e) \mid Ax = b \,;\, x \text{ binary}\}$$

For a given value of λ, L_λ is a minimum cost network flow problem with cost updated as $(c+\lambda.S)$.

Application 5.8 : The Equal Flow Problem

Reference : Ali, Kennington, and Shetty (1987)

The equal flow problem is easily conceptualized as a minimum cost network flow problem with additional constraints on certain pairs of arcs. Specifically, given pairs of arcs are required to take on the same value. The problem is defined on a network presented by an mxn node-arc incidence matrix, A, in which k pairs of arcs are identified and required to have equal flow.

Mathematically, this is expressed as :

[P] min cx

subject to

$$Ax = b \tag{5.8a}$$

$$x_k = x_{k+K}, \quad \forall\, k = 1 \,..\, K, \tag{5.8b}$$

$$0 \le x \le u \tag{5.8c}$$

$$x \text{ integer} \tag{5.8d}$$

where c is a 1xn vector of unit costs, b is an mx1 vector of node requirements, 0 is an nx1 vector of zeroes, x is an nx1 vector of decision variables, and u is an nx1 vector of upper bounds. It is also assumed, without loss of generality, that $u_k = u_{k+K}$ for k = 1,.., K. By the very nature of the constraints (5.8b), these are treated as side constraints.

Applications of equal flow problem are diverse ranging from crew scheduling to estimating driver cost in transit operation to two duty period scheduling problem and so on. The Lagrangian dual of the equal flow problem is obtained by relaxing the equal flow constraints (5.8b) by associating multiplier vector ω with it. The Lagrangian function $h(\omega)$ obtained is as :

$$h(\omega) = \min \{cx + \sum_k \omega_k (x_k - x_{k+K}) \mid Ax = b; 0 \leq x \leq u\}$$

It is obvious that solution to $h(\omega)$ for generating effective lower bounds is obtained by solving a minimum cost network flow problem.

Application 5.9 : Node-Weighted Steiner Tree Problem

Reference : Segev (1987)

This is a natural extension to the Steiner tree problem in which costs associated with node are allowed. It frequently happens in practical solutions that the node weights represent benefits while the edge weight represent costs. Thus, the nodes might represent profit centres and the edges represent the cost required to achieve the profits. This model can be applied to decisions regarding offering of services such as electronic mail and cable TV, where the node weights would represent the revenues obtained from the subscribers. The flow-based formulation of the problem is as follows :

Let f_{ikt} flow of commodity i on arc (k, t)

$$Y_i = \begin{cases} 1 & \text{if node i is in the optimal tree} \\ 0 & \text{otherwise} \end{cases}$$

$$Y_{ij} = \begin{cases} 1 & \text{if arc (i j) is in the optimal tree} \\ 0 & \text{otherwise} \end{cases}$$

The model is as :

$$[P] \quad \text{Min} \sum_{i \in T} C_i X_i + \sum_{i \in T} \sum_{j \in N} C_{ij} Y_{ij}$$

subject to

$$\sum_{t \in T} f_{ikt} - \sum_{t \in N} f_{itk} = 0, \qquad i \in T,\ k \in N,\ k \neq n,\ K \neq 1 \qquad (5.9a)$$

$$\sum_{t \in T} f_{ikt} - \sum_{t \in N} f_{itk} = X_i \qquad i \in T, k = 1 \tag{5.9b}$$

$$\sum_{t \in T} f_{ikt} - \sum_{t \in N} f_{itk} = -X_i \qquad i \in T, k = n \tag{5.9c}$$

$$f_{ikt} \leq Y_{kt} \qquad i \in T, k \in T, t \in T \tag{5.9d}$$

$$Y_{ij}, x_i \in \{0, 1\}$$
(5.9e)

N : set of nodes

T : N - {1}

C_i : cost associated with a node $\{C_i < 0\}$

C_{ij} : cost associated with an arc

In the lower bounding procedure, the above problem is augmented by adding several other constraints, and the coupling variables X_i's are taken care of by their "transfer" to an added dummy node. This changes the constraints (5.9b) and (5.9c) into that of a standard network form. The resulting problem thus has several additional constraints, a few of which are relaxed in a Lagrangian fashion. The resulting problem separates into two subproblems, one of which is a multicommodity network flow problem with no interconnection betwen the commodities. Therefore, the problem is separable into $|T|$ single-commodity flow problems, each of which can be solved as a standard minimum cost flow problem.

CHAPTER 6

APPLICATIONS OF ASSIGNMENT PROBLEM

In this chapter, such applications have been reported in which the relaxed Lagrangian subproblem is an assignment problem. We describe the following applications in the chapter.

1. Lagrangian approach to the traveling salesman problem (Balas and Christofides [1981])
2. Maximum collection problem (Kataoka and Morito [1988])
3. Multi-depot vehicle scheduling problem (Cararessi and Gallo [1984])
4. Prize collecting traveling salesman problem (Balas [1989])
5. Scheduling daily operations in a steel rolling problem (Balas [1989])
6. Personnel assignment in the armed forces

Application 6.1 : Lagrangian Approach to Traveling Salesman Problem

Reference : Balas and Christofides (1981)

The traveling salesman problem (TSP), i.e., the problem of finding a minimum-cost tour (or hamiltonian circuit) in a directed graph G = (N, A), can be formulated as follows :

[P] $\min \sum_{i \in N} \sum_{j \in N} c_{ij} x_{ij}$

subject to

$$\sum_{j \in N} x_{ij} = 1 \qquad \forall\ i \in N \tag{6.1a}$$

$$\sum_{i \in N} x_{ij} = 1 \qquad \forall\ j \in N \tag{6.1b}$$

$$x_{ij} \in \{0, 1\} \qquad \forall\ i, j \in N \tag{6.1c}$$

$$x \text{ is a tour} \tag{6.1d}$$

For $(i, j) \in A$, c_{ij} is the cost associated with the arc (i, j).

Condition (6.1d) can be replaced by the typical subtour elimination constraint as :

$$\sum_{i \in S} \sum_{j \in S} x_{ij} \leq |S| - 1 \qquad S \subseteq N, 1 < |S| < n \tag{6.1e}$$

If we write constraint (6.1e) in its generic form, it becomes :

$$\sum_{i \in N} \sum_{j \in N} a^t_{ij}\, x_{ij} \geq a^t_0 \qquad \forall\ t \in T \tag{6.1f}$$

where S and T are node sets such that

$$(S, T) = \{(i, j) \in A \mid i \in S, j \in T\}$$

TSP is classical optimization problem with wide applicability to the modeling of such important application situations as scheduling, sequencing, routing, etc.

The Lagrangian problem is obtained by relaxing (6.1f) and associating multipliers w :

$$\max_{w \geq 0} L(w)$$

subject to

(6.1a) - (6.1c)

where $L(w) = \min_{x \in X} \sum_i \sum_j (c_{ij} - \sum_{t \in T} w_t a_{ij}^t) X_{ij}) + \sum_{t \in T} w_t a_0^t$

This clearly calls for solving assignment subproblem embedded in constraints (6.1a) - (6.1c) and generates tight lower bounds for the original problem.

Application 6.2 : Maximum Collection Problem

Reference : Kataoka and Morito (1988)

Travelling Salesman Problem (TSP), Vehicle Routing Problem (VRP) and some other studies consider an optimal tour of a graph, assuming that "all" of the nodes of a given graph should be visited exactly once or at least once. This assumption is relaxed in the maximum collection problem, which is stated as :

"Given items with known values located at nodes of network, one wants to collect items so that their total value is maximized under the assumption that a tour starting from the central node and returning to it is completed within predetermined time limits."

Consider an asymmetric graph G (V, E), where V and E are node and arc sets, respectively. $C_i \geq 0$ and $t_{ij} \geq 0$ denote value allocated to node $i \in V$ and time required to travel arc $(i, j) \in E$, respectively. x_{ij} and y_i are both 0-1 variables associated with arc (i, j) and node i, respectively. Let $\bar{V}$ be the set of nodes selected to visit and let S and $\bar{S}$ be the partition of $\bar{V}$ such that $s \cup \bar{S} = \bar{V}$, $S \cap \bar{S} = \phi$, $S \neq \phi$ and $S \neq \bar{V}$. Then, a naive formulation could be presented as :

(FORMULATION 1)

$$\text{Max } \Sigma_{i=1}^{n} c_i y_i$$

subject to

$$\Sigma_{i=1}^{n} \Sigma_{j=1,\, j \neq i}^{n} t_{ij} x_{ij} \leq TC \qquad (6.2a)$$

$$\Sigma_{j=1}^{n} x_{ij} = y_i \qquad \forall\, i \qquad (6.2b)$$

$$\Sigma_{i=1}^{n} x_{ij} = y_j \qquad \forall\ j \tag{6.2c}$$

$$\Sigma_{\,i\in S} \Sigma_{\,j\in \bar{S}} x_{ij} \geq 1 \qquad \text{for any partition of } \hat{V}, (S, \bar{S}) \tag{6.2d}$$

$$x_{ij} \in \{0, 1\},\ y_i \in \{0, 1\},\ y_1 = 1 \text{ \{central node\}} \tag{6.2e}$$

where (6.2a) is the constraint on total time (TC) and (6.2d) is the constraint for subtour elimination. A relaxed problem without these two constraints (with appropriate Lagrangian multipliers) is the assignment problem and thus can be solved easily. However, variables y_i must be determined prior to doing this. The Lagrange multipliers appear only as coefficients of x_{ij} and thus coefficients of y_i are not affected. To alleviate this difficulty, an alternative formulation (2) is contemplated with following transformations :

(Transformation 1) Let ($c_{ij} > 0$) be the cost coefficient of each arc in E where $c_{ij} = c_j$, $i \in V - \{j\}$.

(Transformation 2) Create a self-loop for each node i with $x_{ii} \in \{0, 1\}$ where $c_{ii} = 0$, $t_{ii} = 0$, $i \in V - \{1\}$. For the centre node $c_{11} = 0$, $t_{11} = Tc$.

Now, the alternative formulation could be presented as :

(FORMULATION 2)

$$\text{Max } \Sigma_{i=1}^{n} \Sigma_{j=1}^{n} c_{ij} x_{ij}$$

subject to

$$\Sigma_{i=1}^{n} \Sigma_{j=1,\, j\neq i}^{n} t_{ij} x_{ij} \leq TC \tag{6.2f}$$

$$\Sigma_{j=1}^{n} x_{ij} = 1 \qquad \forall\ i \tag{6.2g}$$

$$\Sigma_{i=1}^{n} x_{ij} = 1 \qquad \forall\ j \tag{6.2h}$$

$$\Sigma_{\,i\in S} \Sigma_{\,j\in \bar{S}} x_{ij} \geq 1 \qquad \text{for any partition } (S, \bar{S}) \text{ of } \hat{V}, \tag{6.2i}$$

$$x_{ij} \in \{0, 1\} \tag{6.2j}$$

Now, the appropriate Lagrangian multipliers could be associated with constraints (6.2f) and (6.2i), respectively and the remaining problem can be solved as a standard assignment problem to generate tight upper bounds.

Application 6.3 : Multi-depot Vehicle Scheduling Problem

Reference : Cararessii and Gallo (1984)

Vehicle scheduling refers to large class of optimization problems in which vehicles must be assigned to time-tabled trips in such a way that

a) each trip is carried out by one vehicle

b) a given set of constraints is satisfied

c) a properly defined cost function is minimised.

Let us assume that a time-table is given, that is a set $V = \{v_i .. v_n\}$ of trips is given where each trip v_i is defined by a quadrupole (τ_i, l_i, o_i, d_i), where:

τ_i the departure time;

l_i the length of time or duration;

o_i the origin or departure terminal;

d_i the destination or arrival terminal;

The deadheading trips are also allowed, that is, the ones from d_i to o_j and their duration is denoted by δ_{ij}.

The ordered pair (v_i, v_j) is said to be a compatible pair of trips if

$$\tau_i + l_i + \delta_{ij} + \varepsilon = \tau_j \tag{6.3a}$$

where $\varepsilon \geq 0$ is prefixed tolerance parameter, introduced to take possible delays into account. Equality (6.3a) states that it is feasible to have trips v_i and v_j operated in sequence by the same vehicle. The authors propose the basic network models and show that with each vehicle scheduling problem we can associate a simple and extremely useful graph. Let $G = \{S, T, A\}$ be a bipartite graph where $S = \{S_i .. S_n\}$ and $T = \{t_i .. t_n\}$ are the sets of nodes and $A = \{(S_i, t_j) : (v_i, v_j)$ is a compatible pair of trips$\}$

is the set of arcs. A matching on G is a subset of arcs $\bar{A} \subseteq A$ such that each node is incident to at most only arc of $\bar{A}$. It is shown that from the bipartite graph (S, T, A), a simple network model can easily be derived, the feasible flow on which defines an assignment between S and T. The most frequent case with vehicle scheduling problems is that of single depot problem, i.e., all vehicles are housed at the same depot. However, with large companies, the problem takes the form of multi-depot, each with a given capacity of housing maximum number of vehicle. The flow model is of multi-commodity type and is much more complex than single-depot problem.

A complex mathematical model is presented in which Lagrangian relaxation coupled with subgradient optimization can be used to generate effective lower bounds.

[P] $\quad \text{Min } \sum_{k=1}^{l} \sum_{(i,j)\in A\cup A^*} c_{ij}\, x_{ij}^k$

subject to

$$\sum_{j\in J(i)} x_{ij}^k = y_{ik} \qquad \forall\, i = 1 .. n,\ k = 1 .. l \tag{6.3a}$$

$$\sum_{i\in I(j)} x_{ij}^k = y_{jk} \qquad \forall\, j = 1 .. n,\ k = 1 .. l \tag{6.3b}$$

$$\sum_{(i,j)\in A^*} x_{ij}^k \le a_k \qquad \forall\, k = 1 .. l \tag{6.3c}$$

$$\sum_{k=1}^{l} y_{ik} = 1 \qquad \forall\, i = 1 .. n \tag{6.3d}$$

$$x_{ij}^k \ge 0 \qquad \forall\, (i, j)\in A\cup A^*,\ k = 1 .. l \tag{6.3e}$$

$$y_{ik} \in (0, 1) \qquad \forall\, i = 1 .. n,\ k = 1 .. l \tag{6.3f}$$

Here

l number of depots

a_k capacity of depot k

A   set of all compatible trips

A^* $= \{(S_i, t_j) : (S_i, t_j) \notin A$ and either $\tau_i \ge \tau_j + l_{ij} + \delta_{ij}$ or $i = j\}$.

$c_{ij} =$ {the cost of deadheading traveling and idle time between the end of trip v_i and the beginning of trips v_j, if $(S_i, t_j) \in A$;

the cost of deadheading from the depot to the departure terminal of v_j plus the cost of deadheading trip from the arrival terminal of v to the depot if $(S_i, t_j) \in A^*$.}

-for $(i, j) \in A$

$$x_{ij}^k = \begin{cases} 1 & \text{if a vehicle housed at depot k runs trips } v_i \text{ and } v_j \text{ in sequence} \\ 0 & \text{otherwise} \end{cases}$$

-for $(i, j) \in A^*$

$$x_{ij}^k = \begin{cases} 1 & \text{if trips } v_i \text{ and } v_j \text{ are the first and the last trips} \\ & \text{respectively of vehicle duties assigned to depot k} \\ 0 & \text{otherwise} \end{cases}$$

-for $i = 1 .. n$

$$y_{ik} = \begin{cases} 1 & \text{if trip } v_i \text{ is assigned to a vehicle housed at depot k} \\ 0 & \text{otherwise} \end{cases}$$

The flow model is of the multicommodity type for the various depots $k = 1, \ldots, l$. It is proposed that there is no need to impose the integrality constraint on x_{ij}^k; since the y_{ik} are fixed (at binary values), the problem decomposes into l smaller assignment problems, and this is where the possibility of using Lagrangian relaxation arises. Although not shown formally through the mathematical model, it is proposed that if we relax constraint number (6.3d), the one binding for all depots, and bring it into the objective function with suitable multipliers, the relaxed problem is separable into l smaller generalized assignment problems.

Application 6.4 : Prize Collecting Travelling Salesman Problem

Reference : Balas (1989)

This problem is a slight variant of maximum collection problem discussed in Application 6.2. The statement of the problem is as follows.

A travelling salesman who gets a prize w_k in every city k that he visits and pays a penalty for every city he fails to visit, and who travels between cities i and j at

the cost c_{ij}, wants to minimize the sum of his travel cost and net penalties, while including in his tour enough cities to collect a prescribed amount w_0 of 'prize money'.

Let y_i be 1 if city i is included in the tour and 0 otherwise, and let x be the incidence vector of the tour, then the problem can be formulated on a complete directed graph G' = (N, A) as

$$\min \quad \Sigma_{i\in N} \Sigma_{j\in N-\{i\}} c_{ij} x_{ij} + \Sigma_{i\in N} p_i (1 - y_i)$$

subject to

$$\Sigma_{j\in N-\{i\}} x_{ij} = y_i \qquad \forall\, i = 1 .. n \tag{6.4a}$$

$$\Sigma_{i\in N-\{j\}} x_{ij} = y_j \qquad \forall\, j = 1 .. n \tag{6.4b}$$

$$\Sigma_{i\in N} w_i y_i \geq w_0 \tag{6.4c}$$

$$y_i \in \{0, 1\} \qquad i\in N,\ x_{ij} \in \{0, 1\},\ (i, j)\in A \tag{6.4d}$$

$$G'\ (y, x) \text{ is a cycle.} \tag{6.4e}$$

where G' (y, x) is the subgraph of G' whose nodes and arcs are those defined by y and x, respectively.

It is convenient for the sake of formulation to complement the variables y_i, $i\in N$, i.e., introduce n new variables $x_{ii} = 1 - y_i$, $i\in N$, to represent the self-loops of graph G = (N, A∪O) obtained from G' by endowing every node with a loop (O is the set of loops). Now, if we define $c_{ii} = p_i$, $i\in N$ and $U = \Sigma_{i\in N} w_i - w_0$, then the problem can be restated as :

$$\min \quad \Sigma_{i=1}^{n} \Sigma_{j=1}^{n} c_{ij} x_{ij}$$

subject to

$$\Sigma_{j=1}^{n} x_{ij} = 1 \quad \forall\, i = 1 .. n \tag{6.4f}$$

$$\Sigma_{i=1}^{n} x_{ij} = 1 \quad \forall\, j = 1 .. n \tag{6.4g}$$

$$\Sigma_{i=1}^{n} w_i x_{ii} \leq U \tag{6.4h}$$

$x_{ij} \in \{0, 1\}$, $\quad\quad i, j = 1 .. n$ (6.4i)

G(x) has exactly one cycle of length ≥ 2 (6.4j)

In this problem, if we relax constraints (6.4h) and (6.4j) in a Lagrangian fashion, the resulting problem is an assignment problem that can generate effective lower bounds to the original problem.

Application 6.5 : Prize Collecting Travelling Salesman Problem : An application to scheduling daily operations in a steel rolling mill

Reference : Balas (1989)

The prize collecting travelling salesman problem can be formulated to model the scheduling the daily operations of a steel rolling mill. The problem is defined as follows :

A rolling mill produces steel sheet from slabs by hot or cold rolling. For reasons that have to do with the wear and tear of the rolls as well as other factors, the sequence in which various orders are processed emerges as a critical factor. Scheduling a round consists of choosing from an inventory of slabs assigned to orders, a collection that satisfies the lower bound on total weight, and ordering it into an appropriate sequence, e.g., one that minimizes some function of sequences. Since the choice of slabs for the round limits the options available for their sequencing, the two tasks must be solved jointly. If a slab for an order is selected and is processed, it gives some benefit and the order which is not processed or is delayed induces some penalty. A particular sequence of order, i.e., one after another, also induces some cost and there could be a constraint that a minimum benefit (or profit) has to be made. From the above description of the problem, it is obvious that Prize Collecting Travelling Salesman Problem as a model captures the essential features of this problem and therefore, the techniques described in the Application 6.4 can be effectively used for this problem as well.

Application 6.6 : Personnel Assignment in Armed Forces

In the armed forces, many men and women are qualified to perform specific jobs, or postings. The armed forces would like to assign the service personnel to postings in such a way that moving costs do not cross a certain limit. The

assignment done correctly in this way result in a "value" for each of such assignment and the objective for armed forces would be to maximize the total value of such assignments. Let d_{ij} be the expected value of an assignment (i, j) and let a_{ij} be the associated posting costs. General rules specify the needed qualifications of the personnel for the postings and identify the jobs that need to be filled. Policy rules determine the allowable assignments that reflect job qualifications and personnel requirements. For an allowable assignment, the posting cost a_{ij} is the dollar cost of moving the person, his or her family, and his or her belongings to the new residence. Let d denote the budget amount available in a period for total posting costs. Then the problem [P] of personnel assignment could be formulated as follows.

$$\min \quad \Sigma_{i=1}^{n} \Sigma_{j=1}^{n} d_{ij} x_{ij}$$

subject to

$$\Sigma_{j=1}^{n} x_{ij} = 1 \qquad \forall\, i = 1 .. n \qquad (6.6a)$$

$$\Sigma_{i=1}^{n} x_{ij} = 1 \qquad \forall\, j = 1 .. n \qquad (6.6b)$$

$$\Sigma_{i=1}^{n} \Sigma_{j=1}^{n} a_{ij} x_{ij} \leq d \qquad (6.6c)$$

$$x_{ij} \in \{0, 1\}, \qquad \forall\, (i, j) = 1 .. n \qquad (6.6d)$$

Here G (N, A) is a graph, with set N of n nodes and set A of m arcs, associated with the assignment problem and x_{ij} is the binary variable denoting whether or not person i is assigned to job j. Constraint (6.6c) ensures that the posting costs remain below a particular amound 'd'.

The problem is easily solved as an assignment problem, if the constraint (6.6c) is relaxed by associating a Lagrangian multiplier $\mu \geq 0$ with it. The relaxed problem is, as follows :

$$\max \quad \Sigma_{i=1}^{n} \Sigma_{j=1}^{n} (d_{ij} + \mu a_{ij}) x_{ij} - \mu d$$

subject to constraints (6.6a), (6.6b) and (6.6d).

CHAPTER 7

ADDITIONAL APPLICATIONS

In this chapter, we report applications whose Lagrangian relaxation result in more than one Lagrangian subproblems, one or more of which may be standard network flow problems. In addition to these, we also report such applications that have not been covered under the previous four chapters. These applications are as follows :

1. Designing optimal railroad operating plans
   (Keaton [1989])
2. Delivery problem
   (Altinkemer and Gavish [1991])
3. Capacitated fixed-charge minimum cost network flow problem
   (Khang and Fujiwara [1991])
4. Steiner problem in graphs
   (Beasley [1984])
5. Star-star concentrator location problem (SSCLP)
   (Mirzaian [1985])
6. Routing in point-to point delivery systems
   (Leung, Magnanti and Singhal [1990])
7. Flow problem with fixed charge : Routing electronic data
   (Hochbaum and Segev [1989])
8. Application of Lagrangian decomposition to resource constrained minimum weight arborescence problem
   (Guignard and Rosenwein [1990])
9. Designing minimal spanning tree subject to a budget constraint
   (Jornsten and Migdalas [1988])
10. Routing for circuit data networks
    (Yee and Lin [1992]).

Application 7.1 : Designing Optimal Railroad Operating Plans

Reference : Keaton (1989)

A railroad can be thought of as a network connecting of terminals which send freight cars to other terminals in the system. Certain pairs of terminals are physically connected by lines of track, as shown in Figure 7.1a.

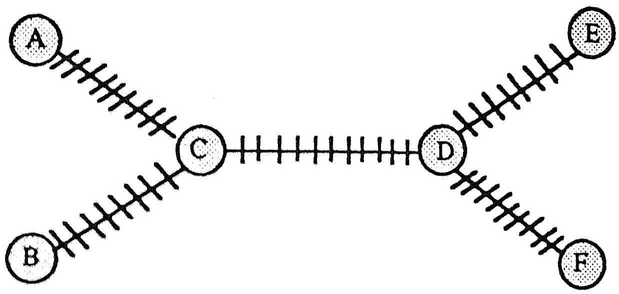


Fig. 7.1a. Rail network.

There is a fixed cost associated with operating a train. There is also a variable cost associated that increases with train size. Because of the level of the fixed costs, railroad managers have a powerful incentive to operate long trains. It is not generally economical to provide all pairs of terminals with direct train connections, and thus many cars must change trains in intermediate terminals, and will encounter a considerable amount of delay in the process. The operating plan problem, therefore, in essence, is to determine :

1) which pair of terminals are to be provided with direct train connections.

2) whether more than one daily train should be operated.

3) how the individual cars are to be routed through the available configuration of trains and intermediate terminals.

The objective is to minimize the fixed train costs, and car time costs.

A network structure is used to represent all the combinations of train connections and blocking (grouping) alternatives which can be used to move cars from origins to destinations. A separate network is required for each origin-destination (O-D) pair. The arcs in the network correspond to 1) the available train connection and 2) the interchange of cars between trains in intermediate terminals.

Let us consider for example cars moving from origin yard A to desitnation yard D in the rail system of Figure 7.1a. A possible car flow network for this O-D pair is shown in Figure 7.1b.

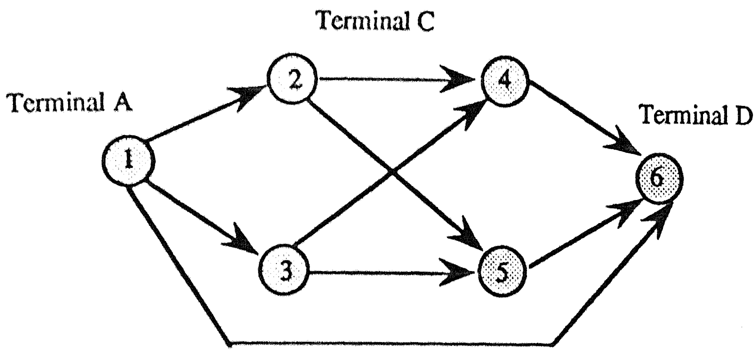


Fig. 7.1b. Car flow network for origin-destination pair (A-D).

Arc (1, 2) corresponds to one daily train from A to intermediate terminal C. Arcs (2, 4), (2, 5), (3, 5), and (3, 4) represent interchange of cars at intermediate terminals. Similarly, arc (1, 6) corresponds to one daily direct train from origin yard A to destination yard D. The other arcs also have similar interpretations.

The model is as follows :

The objective is to determine optimal train connection and car flows so as to minimize car and train costs. Hence, the formulation :

$$\text{Min } cx + dt$$

subject to

$$Ax = b \tag{7.1a}$$

$$x - rt \leq 0 \tag{7.1b}$$

$$t \in S \tag{7.1c}$$

Here, x is a vector of decision variables corresponding to the arcs available to O-D pair, and c is the vector of associated costs. Similarly, t represents the vector of 0-1 decision variables corresponding to the potential train connections in the system, and d is a vector of associated costs.

Constraint (7.1a) represent the allowable car flow network, where A is the node-arc incidence matrix for O-D pair, and b is the number of cars to be moved. Constraint (7.1b) presents cars being assigned to nonavailable trains (say jth for which $t_j = 0$). Here x is the car flow variable for O-D pair associated with its train variable (say x_j for train j) and r is the number of cars moved between a particular O-D pair. Finally, (7.1c) is the relevant constraint on the train variables in set notation.

The solution strategy using Lagrangian Relaxation is as follows :

The Lagrangian relaxation is obtained by "dualizing" the set of constraints $x - rt \leq 0$ which link the train variables and car flow variables. Let us associate a vector of non-negative multipliers u with it and we get the objective function as :

$$\min cx + dt + u(x - rt)$$

Rearranging the terms, we get :

$$Z(u) = \min (c+u)x + (d-r)t$$

subject to

$$Ax = b \tag{7.1d}$$

$$t \in S \tag{7.1e}$$

This relaxed problem is easy to solve, since it decomposes into two separate problems, one in the x-variables and the other in t-variables.

The x-variable problem :

$$\min (c+u)x$$

subject to

$$Ax = b$$

is simply a series of shortest path problems, one for each O-D pair, with costs of (c+u) on arcs. The author also show that solving the other half of the relaxed problem is not computationally burdensome. We can proceed with the x-variable problem using the subgradient optimization technique and can obtain good lower bounds.

Application 7.2 : Delivery Problem

Reference : Altinkemer and Vavish (1991)

The delivery problem is one of a group of closely related problems called the vehicle routing problem (VRP). It looks for the most economical way to fulfill the deterministic transportation requirements of customers by a fleet of vehicles. The requirements of customers could be pickup or delivery. Real life applications of the delivery problem include delivery of products to supermarkets and departmental stores, picking up students by school buses, distribution of newspaper mail or laundry and the topological design of local access ring networks.

Each vehicle tour starts and terminates at the central depot, and the load to be delivered at each node is supplied by exactly one vehicle. A set of nodes that is on the same tour (i.e., served by the same vehicle) form a cluster. The total delivery cost includes operating expenses, fuel and labor. The mathematical formulation is presented below :

n the number of nodes in the problem;

V the node set of the graph, $V = \{1 .. n\}$;

$\check{V}$ the node set of the graph excluding root (central depot);

E the edge set of the graph

Q the vehicle capacity

d_i the load to be delivered to node i ($d_i = 0$)

c_{ij} the cost of travel from node i to node j.

Decision variable $x_{ij} = \begin{cases} 1 & \text{if arc (i j) is traveresed by a vehicle} \\ 0 & \text{otherwise} \end{cases}$

Model

$$\text{Minimize } Z_{DP} = \Sigma_{i=1}^{n-1} \Sigma_{j=i+1}^{n} c_{ij} x_{ij} + \Sigma_{i=2}^{n} c_{i1} x_{i1}$$

subject to

$$x_{j1} + \Sigma_{i=1}^{j-1} x_{ij} + \Sigma_{i=j+1}^{n} x_{ji} = 2 \qquad \forall\, j = 2 \,..\, n\text{-}1 \tag{7.2a}$$

$$x_{n1} + \Sigma_{j=1}^{n-1} x_{jn} = 2 \tag{7.2b}$$

$$\Sigma_{i=2}^{n} x_{j1} \geq L_{\bar{V}} \tag{7.2c}$$

$$\Sigma_{i=1}^{n-1} \Sigma_{j=i+1}^{n} x_{ij} = n\text{-}1 \tag{7.2d}$$

$$\Sigma_{i\in S} \Sigma_{i<j\in S} x_{ij} \leq |S| - L_S \text{ for all } S \subseteq \{2 \,..\, n\} \tag{7.2e}$$

$$x_{ij} \in \{0, 1\}, \qquad \text{for all } i, j \tag{7.2f}$$

The first and second set of constraints ensure that exactly one vehicle leaves and enters from every delivery node. The third constraint set provides a lower bound on the number of returning arcs to the central depot and is redundant. However, it will be helpful in the Lagrangian relaxation to tighten the lower bound on the optimal solution value. L_S is the optimal solution to the bin-packing problem (Eilon and Christofides [1971]), when the bin length is Q and D_i; $i \in S$ are the lengths of items to be packed. Constraint (7.2d) follows from the fact that any feasible solution must have (n-1) arcs in addition to the arcs directly returning to the central depot. Constraint (7.2e) is the subtour elimination constraint and the use of this particular form of constraint is made in Lagrangian relaxation. For this, constraint set (7.2e) is divided into two subsets

$$\Sigma_{i\in S} \Sigma_{i<j\in S} x_{ij} \leq |S| - 1$$

$$\text{for all } S \subseteq \{2 \,..\, n\},\ L_S = 1,\ |S| \geq 2, \tag{7.2ea}$$

$$\Sigma_{i\in S} \Sigma_{i<j\in S} x_{ij} \leq |S| - L_S$$

$$\text{for all } S \subseteq \{2 \,..\, n\},\ L_S \geq 2,\ |S| \geq 2, \tag{7.2eb}$$

The Lagrangian relaxation $L(\alpha, \pi)$ is generated by multiplying constraints (7.2a), (7.2b) and (7.2eb) with multipliers $\alpha = \{\alpha_i \,..\, \alpha_n\}$ and $\pi = \{\pi_i \,..\, \pi_n\}$, α unrestricted and $\pi \geq 0$.

$$L(\alpha, \pi) = \Sigma_{i=1}^{n-1} \Sigma_{j=i+1}^{n} c_{ij} x_{ij} + \Sigma_{i=2}^{n} c_{i1} x_{i1} + \Sigma_{S\in\varphi} \Sigma_{i<j\in S} x_{ij} \leq (|S| - L_S)\pi_S$$

$$- \Sigma_{S\in\varphi} \Sigma_{i\in S} \Sigma_{i<j\in S} \pi_S x_{ij} + \Sigma_{j=2}^{n-1} \alpha_j \left(2 - x_{j1} - \Sigma_{i=1}^{j-1} x_{ij} - \Sigma_{i=j+1}^{n} x_{ji}\right)$$

$$+ \alpha_n (2 - x_{n1} - \Sigma_{j=1}^{n-1} x_{jn})$$

subject to constraints (7.2c), (7.2d), (7.2ea), and (7.2f), where φ represents the set of unordered subsets of {2 .. n} with a cardinality greater than 1, such that each subset requires more than one vehicle (bin). For a given set of multipliers π and α, $L(\alpha, \pi)$ is a degree constrained minimal spanning tree (DCMST). Now, one can proceed with subgradient optimisation technique to generate good lower bounds for the delivery problem.

Application 7.3 : Capacitated Fixed-charge Minimum Cost Network Flow Problem

Reference : Khang and Fujiwara (1991)

The capacitated fixed-charge minimum cost network flow problem arises in a variety of real-world applications. The presence of a fixed cost of using an arc in a network, in addition to variable cost per unit of flow in the arc, is a familiar phenomenon in many transportation, distribution and communication network systems. The reference for such applications has been made to the paper by Magnanti and Wong (1984), wherein a lot of applications of fixed charge network problems as travelling salesman problem, vehicle routing problem, facility location problem, network design and traffic equilibrium can be found.

Let us consider the model formulation of fixed-charge network flow problem. Let G (N, A) be a connected and directed graph with node set N = {1 ,.., n} and the arc set A. Suppose that each arc (i, j) in A is assigned a fixed charge (nonnegative) s_{ij}, a variable cost per unit c_{ij}, and a positive capacity b_{ij}. For a node i in N, let r_i be the fixed requirement at i. The capacitated, single-commodity, fixed-charge network flow problem (P) is :

$$\text{minimize } f(x) = \sum_{(i,j)\in A} c_{ij}\, x_{ij} + \sum_{(i,j)\in A} s_{ij}\, \delta(x_{ij})$$

subject to

$$\sum_{(k,i)\in A} x_{ki} - \sum_{(i,j)\in A} x_{ij} = r_i \quad \forall\, i = 2 ,.., n \tag{7.3a}$$

$$0 \le x_{ij} \le b_{ij} \qquad \forall (i, j)\in A \tag{7.3b}$$

where x_{ij} = flow on arc (i, j)

$$\sum_{i \in N} r_i = 0, \quad \delta(z) = \begin{cases} 1 & \text{if } x > 0 \\ 0 & \text{otherwise} \end{cases} \tag{7.3c}$$

It is assumed that the network has a single supply node (denoted as node 1) and that all other nodes have positive demands, i.e., $r_1 < 0$ and $r_i > 0$ for $i = 2, ..., n$. Before considering the relaxation scheme, let us exhibit an equivalent formulation of P, denoted by Q :

$$\text{minimize } g(x, y, z) = \sum_{(i,j) \in A} c_{ij} x_{ij} + \sum_{(i,j) \in A} s_{ij} y_{ij}$$

subject to

$$\sum_{(k,j) \in A} x_{ki} - \sum_{(i,j) \in A} x_{ij} = r_i \quad \forall\, i = 2 .. n \tag{7.3d}$$

$$b_{ij} z_{ij} \leq x_{ij} \leq b_{ij} y_{ij} \quad \forall\, (i, j) \in A \tag{7.3e}$$

$$y_{ij}, z_{ij} \in \{0, 1\} \quad \forall\, (i, j) \in A \tag{7.3f}$$

The equivalence of P and Q is obtained by choosing $y_{ij} = \delta(x_{ij})$ and $z_{ij} = 1 - \delta(b_{ij} - x_{ij})$ for any feasible solution x_{ij} of P.

Now, let us denote by u_i $(i = 2, ..., n)$ the Lagrange multipliers, unrestricted in sign, corresponding to constraint (7.3d). Then, the relaxed problem is as follows :

$$\text{Minimize} \quad \sum_{(i,j) \in A} (c_{ij} x_{ij} + s_{ij} y_{ij}) + \Sigma_{i=2}^{n} u_i \Big(\sum_{(k,j) \in A} x_{ki} - \sum_{(i,j) \in A} x_{ij} - r_i \Big)$$

subject to

constraints (7.3e) and (7.3f)

The authors show that after several modifications of the above, the relaxed problem is equivalent to an MST (minimum spanning tree) problem on G, with appropriately defined weights.

Application 7.4 : Steiner Problem in Graphs

Reference : Beasley (1984)

It is the problem of connecting together at minimum cost a set of vertices in an undirected graph. This problem is NP-complete and therefore tree search procedure based on Lagrangian Relaxation is used. The mathematical formulation is as :

Let

V set of entire vertices

k set of vertices to be connected together, $k \subseteq V$

E set of (undirected) edges

c_{ij} ≥ 0, cost of edge $(i, j) \in E$

The problem is to choose a subset of E which connects together all the vertices in k at minimum cost.

Let $k_1 = k-1$

$$y_{ij} = \begin{cases} 1 & \text{if edge (i j) is in the solution} \\ 0 & \text{otherwise} \end{cases}$$

$$x_{ijk} = \begin{cases} 1 & \text{if edge (i j) is in the unique path in solution tree from 1 to k} \\ 0 & \text{otherwise} \end{cases}$$

Then, the program is

[P] Minimize $\sum_{(i,j) \in E} c_{ij} \, y_{ij}$

subject to

$$y_{ij} \geq x_{ijk} \qquad \forall \, k \in K_1, \ \forall \, (i, j) \in E \tag{7.4a}$$

$$P_1 \leq \sum_{(i,j) \in E} y_{ij} \leq P_u \tag{7.4b}$$

there exists an elementary path (x_{ijk}) from 1 to k $\forall \, k \in K_1$ (7.4c)

$$x_{ijk} \in (0, 1) \qquad \forall \, k \in K_1, \forall \, (i, j) \in E \tag{7.4d}$$

$y_{ij} \in (0, 1) \qquad \forall\, (i, j) \in E \qquad (7.4e)$

Constraint (7.4a) ensures that an edge is only on some path in the tree if that edge is in the solution tree. Constraint (7.4b) limits the number of edges in the solution tree, where $P_1 = |K| - 1$, $P_u = |V| - 1$. Constraint (7.4c) ensures that the solution is connected. To develop the lower bound for the problem [P], we relax equation (7.4a) in a Lagrangian fashion. Let s_{ijk} be the associated multipliers. Then, the relaxed problem (P_R) is :

$[P_R]$ Minimize

$$\sum_{(i,j)\in E} (c_{ij} - \sum_{k\in K_1} s_{ijk})\, y_{ij} + \sum_{k\in K_1} \sum_{(i,j)\in E} s_{ijk}\, x_{ijk}$$

subject to (7.4b) - (7.4e)

The Lagrangian problem can be easily solved as it decomposes into two separate problems, the solution consisting of (i) the calculation in the variables y_i's, and (ii) $|K_1|$ shortest elementary path calculations, finding the shortest path from vertex 1 to vertex $k \in K_1$ inthe graph with cost matrix s_{ijk}.

Application 7.5 : Star-Star Concentrator Location Problem (SSCLP)

Reference : Mirzaian (1985)

It is a network layout problem and is stated as follows. An important communication design problem is how to connect several remote terminal sites T_i, $i = 1, \ldots, n$ to a central (processing) site C_o. The usual design method uses concentrators. We are given c_j, $j = 1, \ldots, m$, a set of potential concentrator sites which is usually a subset of terminal sites, and a subset $Y \subseteq \{c_1, \ldots, c_m\}$ is selected to be the set of actual concentrator sites. These concentrators will be connected to the central site via high-speed lines. Each terminal must be connected via a low-speed line, to a unique site in $Y \subseteq \{c_o\}$. The number of terminals connected to any $c_j \in Y$ must not exceed a positive integer $k (\leq n)$ called the concentrator capacity. The cost of installing a concentrator at site c_j and connecting it to the central site through a high-speed line is d_j. The cost of connecting terminal T_i to c_j is c_{ij}. The optimization problem to find a network with minimum cost is called the star-star concentrator location problem

(SSCLP). Having defined the problem in this way, the mathematical formulation is very easy to state.

Let, x_{ij} be a binary 0-1 variable denoting whether or not T_i is connected to c_j. Similarly, y_j is a binary 0-1 variable denoting whether or not c_j is selected as a concentrator site. Then, SSCLP is :

[P] $\quad$ min. $\Sigma_{i=1}^{n} \Sigma_{j=0}^{m} c_{ij} x_{ij} + \Sigma_{j=1}^{n} d_j y_j$

subject to

$$\Sigma_{j=0}^{m} x_{ij} = 1 \qquad \forall\, i = 1 .. n \tag{7.5a}$$

$$\Sigma_{i=1}^{n} x_{ij} \le k y_j \qquad \forall\, j = 1 .. m \tag{7.5b}$$

$$x_{ij}, y_j \in \{0, 1\} \tag{7.5c}$$

The author proposes two Lagrangian relaxations of the problem, by relaxing (7.5a) and (7.5b), respectively. The relaxation obtained by relaxing (7.5a) provides tighter lower bounds. Hence, relaxed problem, $LR_1(\lambda)$ in vectorial notation is as :

[$LR_1(\lambda)$] $\quad$ min. $[cx + dy + \Sigma_{i=1}^{n} \lambda_i (1 - \Sigma_{j=1}^{m} x_{ij})]$

$$= \min [\, \Sigma_{i=1}^{n} \Sigma_{j=0}^{m} (c_{ij} - \lambda_i) x_{ij} + \Sigma_{j=1}^{m} d_j y_j + \Sigma_{i=1}^{n} \lambda_i \,)$$

subject to (7.5b) and (7.5c) of P.

Next, we state the following result without the proof, which cannot be included fully here due to its complexity. The interested readers may, however, refer to the original paper by Mirzaian.

Observation: The problem $LR_1(\lambda)$ is equivalent to problem $\hat{P}$ which is shown as

[$\hat{P}$] $\quad$ min $\Sigma_{i=1}^{n} \Sigma_{j=0}^{m} \tilde{q}_{ij} x_{ij}$

subject to

$$\Sigma_{j=0}^{m} x_{ij} = 1 \qquad \forall\, i = 1 .. n \tag{7.5d}$$

$\Sigma_{i=1}^{n} x_{ij} \leq k$ $\quad\quad \forall\ j = 1 .. m$ $\quad\quad$ (7.5e)

$x_{ij} \geq 0$ $\quad\quad \forall\ i = 1 .. n, j = 0 .. m$ $\quad\quad$ (7.5f)

In $\hat{P}$, note that the variable y_j's have been eliminated. Further, the author shows that the problem $\hat{P}$ is equivalent to another problem $\tilde{P}$, which is shown as :

$[\tilde{P}]$ $\quad \min \Sigma_{i=1}^{n} \Sigma_{j=0}^{m} \tilde{q}_{ij}\, x_{ij}$

subject to

$\Sigma_{j=0}^{m} x_{ij} = a_i$ $\quad\quad \forall\ i = 1 .. n$ $\quad\quad$ (7.5g)

$\Sigma_{i=1}^{n} x_{ij} = b_j$ $\quad\quad \forall\ j = 1 .. m$ $\quad\quad$ (7.5h)

$x_{ij} \geq 0$ $\quad\quad \forall\ i = 0 .. n, j = 0 .. m$ $\quad\quad$ (7.5i)

where

$$a_i = \begin{cases} 1 & \text{if } i = 1 .. n \\ km & \text{if } i = 0 \end{cases}$$

$$b_j = \begin{cases} k & \text{if } j = 1 .. m \\ n & \text{if } j = 0 \end{cases}$$

Note that the above definition of a_i 's and b_j's is used to balance the supply and demand of two sets of nodes in a transportation problem, after a dummy node has been added to each set, one having a supply of km and the other having a demand of n. Hence, it is shown finally that the relaxed problem $LR_1(\lambda)$ is equivalent to a transportation problem with appropriately defined cost, supply and demand functions.

Application 7.6 : Routing in Point-to-Point Delivery Systems

Reference : Leung, Magnanti and Singhal (1990)

In a wide variety of large scale delivery systems, a firm must ship goods on a network between many origin and destination (O-D) pair of nodes. The firm would like to specify a cost-effective route for each of the O-D pairs in order to minimize overall transportation costs and/or transit times. This type of delivery route

planning problem arises in numerous application settings, such as package delivery, mail delivery, telephone network design, and rail freight shipping. The delivery network consists nodes of two types - terminal nodes and distribution centres (see Figure 7.6).

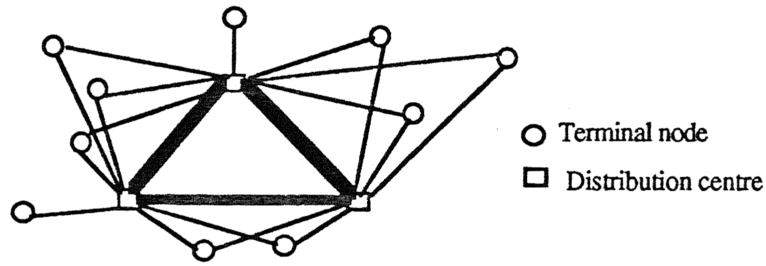


Fig. 7.6. A sample delivery network

The required volume of loads to be shipped O-D pair is specified *a priori* . The system incurs two types of costs - shipping costs on each link of the network and the processing cost at each node. It should be noted that modeling of shipments between O-D pairs has been considered in this paper and not collection/distribtion of goods from/to customers. As an example, in mail delivery, the terminals will correspond to local post offices, and the distribution centers to major sorting facilities. Therefore, it differs from the typical vehicle routing problem, in which the local pickup and delivery is done for end customers. Although there could be many operational issues to be handled in such a scenario, the model handles two of those :- 1) All goods between the same O-D pair use the same route, and 2) Flows must satisfy service requirements, which guarantee delivery within a certain time. The authors propose a complex mathematical model and enough justice cannot be done by including the mathematical formulation here. We rather give a brief description of the model and the solution strategies using Lagrangian relaxation.

The objective is to minimize shipping cost on each link and processing cost at each node. The diferent constraints serve the following purposes. Constraints (1) and (5) force each O-D pair to be assigned to a DC (Distribution Center) - node pair. Constraint (2) ensures that the DC's allocate sufficient capacity to cover the volume processed. Constraints (3), (4), and (10) ensure that the number of trailers dispatched on a link is sufficient to carry the volume on that link. The other constraints take care of the connectedness of the network. This particular formulation reflects the solution approach via decomposition by considering the routing decisions for each O-D pair in two main steps, namely, 1) The assignment of each O-D pair to a (first

DC - last DC) pair, and 2) The choice of a route from the first DC to last DC node. The model separates the original problem into two linked subproblems. The first problem considers the assignment of a first and last DC node on the route for every O-D pair. After all such assignments have been made, the second step seeks a minimum cost routing of the aggregated flow of goods among the DC nodes.

In the assignment subproblem, the use of Lagrangian relaxation is made in the following way. The solution approach to this problem iterates between a trailer allocation problem and an (O-D pair) to (DC pair) assignment problem. First, the maximum number of trailers that can be sent on each (O-D pair) - DC link are decided. Then, using the Lagrangean approach, the cheapest assignment honouring the limit on allowable number of trailers on each link is found out. The trailer allocation on several of the (O-D pair) - DC links is adjusted and a compatible assignment is tried out again. Finally, after repeating these steps several times, the final solution is the one with the minimum overall cost.

Application 7.7 : Flow Problem with Fixed Charges : Routing Electronic Data

Reference : Hochbaum and Segev (1989)

This paper addresses a problem of network design, referred to as a fixed-charge problem. The set up for this problem includes a root node from which flow is to originate, a collection of nodes all of which have various demands, and the directed arcs between all pair of nodes each with two cost figures. One of the costs is the fixed cost of using an arc to send flow and the other is the variable cost per unit of flow on that arc. The objective is to find minimum cost-set of arcs along which the flows from the root to the demand node is to be sent. An application setting is in routing messages of electronic data. Here, the root node sends messages at a certain point in time addressed to a specific collection of nodes. The use of each arc has initial set-up cost asociated with it (such as the surcharge for the first minute of a telephone call) and variable cost per unit of data transmitted. The routing decisions are to be made starting at the root node. Then the details of the routing decisions can be transmitted with the data.

Let there be a directed graph G = (V, E), where V is set of n nodes and E is the set of m arcs. Each arc (i, j) has fixed cost d_{ij} and variable cost c_{ij} associated with it. The graph has one distinguished node V_1 from which all flows originate - the

source. The other (n-1) nodes $V_2, \ldots, V_n$ have "demand" weights $a_2, \ldots, a_n$ associated with them. Let x_{ij} be the amount of flow on arc (i, j) and y_{ij} is a 0-1 variable which is one if $x_{ij} > 0$ and zero, otherwise.

The mathematical formulation is as

[P] min. $\Sigma_{i=1}^{n} \Sigma_{j=2}^{n} c_{ij} x_{ij} + \Sigma_{i=1}^{n} \Sigma_{j=2}^{n} d_{ij} y_{ij}$

subject to

$$\Sigma_{p=1}^{n} x_{pj} - \Sigma_{q=2}^{n} x_{jq} = a_j \quad \forall j = 2, \ldots, n \tag{7.7a}$$

$$\Sigma_{i=1}^{n} y_{ij} = 1 \quad \forall j = 2, \ldots, n \tag{7.7b}$$

$$Ay_{ij} \geq x_{ij} \quad \forall i = 1, \ldots, n, \; j = 2, \ldots, n \tag{7.7c}$$

$$x_{ij} \geq a_j y_{ij} \quad \forall i = 1, \ldots, n, j = 2, \ldots, n \tag{7.7d}$$

$$Aw_{ij} \geq x_{ij} \quad \forall i = 1, \ldots, n, \; j = 2, \ldots, n \tag{7.7e}$$

$$\Sigma_{i=1}^{n} w_{ij} = 1 \quad \forall j = 2, \ldots, n \tag{7.7f}$$

$$\Sigma_{(i,j) \in S} y_{ij} \leq |S| - 1 \quad \forall S \subseteq \{1, .., n\} \tag{7.7g}$$

$$\Sigma_{(i,j) \in S} w_{ij} \leq |S| - 1 \quad \forall S \subseteq \{1, .., n\} \tag{7.7h}$$

$$w_{ij} \in \{0, 1\}, \; y_{ij} \in \{0, 1\}, x_{ij} \geq 0 \quad \forall i = 1, \ldots, n, \; j = 2, \ldots, n \tag{7.7i}$$

Here, $A = \Sigma_{j=2}^{n} a_j$ is total flow originating from source. w_{ij} has the same interpretation as y_{ij}, though it does not appear in the objective function.

Constraints (7.7a) represent flow balance constraint; constraints (7.7b) ensure that exactly one arc will enter each node (except node 1); constraints (7.7c) guarantee that if $y_{ij} = 0$, no flow will occur between node i and node j. This formulation is in fact an augmented formulation of a typical fixed charge problem. The formulation has been augmented by adding a set of redundant constraints. Though redundant in this formulation, the additional constraints (7.7d) through (7.7h) help to generate tighter lower bounds via Lagrangian relaxation. Constraints (7.7g) and (7.7h) are typical cycle prevention constraints. The Lagrangian relaxation is proposed by

relaxing these constraints by associating non-negative multiplier vectors z and λ with them, respectively. Then, we get the relaxed problem as :

$$[P_R] \quad \min \; \{\Sigma_{i=1}^{n} \Sigma_{j=2}^{n} \bar{c}_{ij} x_{ij} + \Sigma_{i=1}^{n} \Sigma_{j=2}^{n} \bar{d}_{ij} y_{ij}\}$$

subject to (7.7a), (7.7b), (7.7e) - (7.7i)

where $\bar{c}_{ij} = c_{ij} + z_{ij} - \lambda_{ij}$ and $\bar{d}_{ij} = d_{ij} + a_j \lambda_{ij} - Az_{ij}$

The relaxed problem has a nice property that it is decomposable into two simpler subproblems. The first is a minimum branching rooted at node 1 (with reference to y_{ij}) and the second problem is a minimum tree-flow problem (with reference to w_{ij} and x_{ij}). The solution of the tree-flow problem is obtained by finding shortest paths fom node 1 to all other nodes.

Application 7.8 : Application of Lagrangian Decomposition to Resource Constrained Minimum Weight Arborescence Problem.

Reference : Guignard and Rosenwein (1990)

This is an extension of the well-known network design problem, and is also known as directed minimum spanning tree problem. It arises frequently in the design of a hierarchial distribution network in which finite resources, pertinent to distribution, exist at each vertex.

An arborescence, rooted at S, is a connected partial graph G' = (V, B) of G = (V, A) where $B \subseteq A$ {V is the set of vertices and B, A are set of arcs}. Let $x_{ij} = 1$ if $x \in B$ and 0, otherwise.. Among the properties characterizing an arborescence are that 1) each vertex except S has an indegree of exactly one; 2) it has (n-1) arcs; 3) it contains no circuits; 4) a unique path exists from s to every $j \in V'$, where V' = V-{s}. If each arc (i, j) has an associated weight, c_{ij}, the weight of the an arborescence B emanating from s is $\sum_{(i,j)\in B} c_{ij}$. The problem $\min_B \sum_{(i,j)\in B} c_{ij}$ represents the minimum weighted arborescence (MWA) problem.

$$[P] \qquad \text{Min} \sum_{(i,j)\in A} c_{ij} x_{ij}$$

subject to

$$\sum_{i \in P(j)} x_{ij} = 1, \qquad \forall \; j \in V' \tag{7.8a}$$

G' contains no circuits (7.8b)

$$\sum_{j \in Q(i)} a_{ij} \, x_{ij} \leq b_i, \qquad \forall \; i \in V \tag{7.8c}$$

$$x_{ij} = 0 \text{ or } 1, \qquad \forall \; (i, j) \in A \tag{7.8d}$$

Here,

P(j)	set of predecessors of node j.
Q(i)	set of successors of i
a_{ij}	amount of resource associated with traversing an arc (i, j)
b_i	fixed amount of resource at each vertex 'i'.

The presence of (7.8c) implies that the problem is NP-hard. Therefore, a redundant constraint (8.8e) $x_{ij} = y_{ij}$ is added to the problem and the decision variables in (7.8c) are reframed from x to y. Associating multiplier vector u with (7.8e) and relaxing it in a Lagrangian fashion yields two subproblems - one an MWA problem and the other a set of n independent knapsack problems.

[P_{R1}]

$$\text{Min}_x \sum_{(i,j) \in A} (c_{ij} + u_{ij}) \, x_{ij}$$

subject to

$$\sum_{i \in P(j)} x_{ij} = 1, \qquad \forall \; j \in V' \tag{7.8e}$$

G' contains no circuits (7.8f)

$$x_{ij} = 0 \text{ or } 1, \qquad \forall \; (i, j) \in A \tag{7.8g}$$

[P_{R2}]

$$\mathrm{Max}_y \sum_{(i,j)\in A} u_{ij}\, y_{ij}$$

subject to

$$\sum_{j\in Q(i)} a_{ij}\, y_{ij} \le b_i, \qquad \forall\ i\in V \tag{7.8h}$$

$$y_{ij} = 0 \text{ or } 1, \qquad \forall\ (i, j)\in A \tag{7.8i}$$

P_{R1} can be solved efficiently as a minimum spanning tree problem, with cost vector as $(c_{ij} + u_{ij})$.

Application 7.9 : Designing Minimal Spanning Tree Network Subject to a Budget Constraint

Reference : Jornsten and Migdalas (1988)

The problem of determining a minimal spanning tree (MST) subject to side constraints arises frequently in practice, generally in the design of computer communication networks, and pipeline systems. The inherent structure of the side constraints may vary for different applications, thereby defining a whole family of constrained minimal spanning tree problems. The well known degree constrained MST, capacitated MST, resource constrained MST, and the MST subject to flow requirements, all belong to this class of problem and can be described in terms of a generic model. The model and the notations are as follows :

Let G (N, A) be a graph with N as set of nodes and A as set of arcs. Let node 1 be distinguished from the rest of the nodes. We assume that every node, except node 1, generates 1 unit of flow. Node 1 is designated as sink. Hence, the problem is to connect N nodes of the network in the most cost-efficient way, while at the same time preserving the tree topology of the solution and satisfying the budget constraint. This problem P_1 can be stated as follows :

$[P_1]$

$$\min_{(i,j)\in A} f_{ij}\, y_{ij} \tag{7.9a}$$

subject to

$$\sum_{j\in N\,(i,j)\in A} x_{ij}^k - \sum_{j\in N\,(i,j)\in A} x_{ji}^k = \begin{cases} 1 & \text{if } i=k \\ -1 & \text{if } i=1 \\ 0 & \text{otherwise} \end{cases} \quad \forall\, i\in N,\ \forall\, k\in N-\{1\}$$

(7.9b)

$$x_{ij}^k \le y_{ij} \qquad \forall\,(i, j)\in A,\ \forall\, k\in N-\{1\} \qquad (7.9c)$$

$$x_{ij}^k \ge 0 \qquad \forall\,(i, j)\in A,\ \forall\, k\in N-\{1\} \qquad (7.9d)$$

$$y_{ij}\in \{0, 1) \qquad \forall\,(i, j)\in A \qquad (7.9e)$$

$$\sum_{(j\in N)((i,j)\in A)} y_{ij}=1 \quad \forall\, i\in N-\{1\} \qquad (7.9f)$$

$$\sum_{(i,j)\in A} a_{ij}\, y_{ij} \le b \qquad (7.9g)$$

Here, f_{ij} is the cost of including the arc (i, j) in the solution. The binary variable y_{ij} represents the choice of whether to include the arc (i, j) in the solution or not, and the continuous variables x represent the flow generated at node k which passes through arc (i, j). The constraint (7.9b) are the well known equations for flow conservation. Constraints (7.9c) imply that an arc is only open for traffic if it has been constructed. The out-degree constraint (7.9f) ensure the tree-like structure and constraint (7.9g) is the budget restriction.

By introducing auxillary variables z_{ij} and $|A|$ constraints $z_{ij} = y_{ij}$, the following problem equivalent to P_1 results :

[P_2]

$$\min \qquad \alpha \sum_{(i,j)\in A} f_{ij}\, y_{ij} + \beta \sum_{(i,j)\in A} f_{ij}\, z_{ij} \qquad (7.9h)$$

subject to (7.9b) - (7.9f)

$$\sum_{(i,j)\in A} a_{ij}\, z_{ij} \le b \qquad (7.9i)$$

$$z_{ij} \in \{0, 1\} \qquad \forall\,(i, j)\in A \qquad (7.9j)$$

$$y_{ij} = z_{ij} \qquad \forall\,(i, j)\in A \qquad (7.9k)$$

Here α and β are prefixed multipliers such that $\alpha + \beta = 1$.

Two Lagrangian relaxation proposals are presented, one each corresponding to both P_1 and P_2.

By relaxing the budget constraint (7.9g), problem P_1 reduces to the following MST-subproblem :

$$v(MST; \mu) = \min\ (f_{ij} + \mu a_{ij}) y_{ij} \tag{7.9l}$$

subject to (7.9b) - (7.9f)

The solution of this problem can easily be obtained by applying any minimum spanning tree algorithm.

Corresponding to P_2, the Lagrangian relaxation of constraints (7.9k), separates the problem into one (x, y)-part and one z-part, resulting in the following subproblem :

$$v(MST; \lambda) = \min \sum_{(i,j) \in A} (\alpha f_{ij} + \lambda_{ij}) y_{ij} \tag{7.9m}$$

subject to (7.9b) - (7.9f)

and

$$v(KNPSK; \lambda) = \min \sum_{(i,j) \in A} (\beta f_{ij} - \lambda_{ij}) z_{ij} \tag{7.9n}$$

subject to

$$\sum_{(i,j) \in A} a_{ij} z_{ij} \le b \tag{7.9p}$$

$$z_{ij} \in \{0, 1\} \qquad \forall\ (i, j) \in A \tag{7.9q}$$

Subproblem (7.9m) is a minimal spanning tree subproblem and subproblem (7.9n) is a zero-one knapsack problem.

Application 7.10 : Routing for Circuit Data Networks

Reference : Yee and Lin (1992)

In virtual circuit networks, the data packets are considered for transmission between a set of origin - destination (O-D) pairs. All the packets in a session are transmitted over exactly one path established between the O-D pair. For each O-D pair, it is assumed that there are multiple sessions. The main motivation for using this type of service is that all the packets arrive at the destination in the order in which they are sent. The problem is to choose a path for each session so as to minimize the packet delay in the network. The problem is formulated as a multicommodity flow problem. Because of the complexity of the mathematical model, it cannot be included here in totality. The scheme of the usage of the Lagrangean relaxation is as follows. The capacity of each link in the network is limited and it represents a hard constraint in the model. When this capacity constraint is relaxed in a Lagrangean fashion, the problem divides in two subproblems. Subproblem 1 is composed of $|L|$ (one for each link l) simple problems. Each of such problems is easily solvable by minimization of a convex function. Subproblem 2 composes of $|W|$ (one for each O-D pair w) independent problems, each involving the rearrangement of the distribution of the sessions among the available route sets. The authors show that the above is equivalent to solving a shortest path problem for each O-D pair. The interested readers are referred to the original paper for more detailed description of the model and the solution procedure by Lagrangean relaxation and subgradient optimization.

CHAPTER 8

CONCLUSIONS AND DIRECTION FOR FUTURE RESEARCH

Through the appreciation of the applications presented in the previous chapters, we observe that Lagrangian relaxation emerges as a flexible solution strategy that permits the modelers to exploit the underlying structure in any optimization problem by relaxing complicating constraints. Usually, such constraints are chosen to relax that furnish a Lagrangian subproblem which is much easier to solve than the original problem.

The applications in this work encompass several important network optimization models: networks with side constraints, the traveling salesman problem, vehicle routing, personnel planning, degree-constrained minimum spanning trees, and production planning, etc. As we have seen, these optimization models have applications in such diverse settings as communication systems design, crew scheduling, logistics, production, workforce scheduling, and multilocation facility modernization, etc. Consequently, this work illustrates the broad applicability of Lagrangian relaxation across many practical problem contexts and demonstrates the ability of network flows to model an extraordinarily wide range of problems met in practice. Therefore, it appears pertinent at this stage to categorize these applications according to the problem context, thus providing a reference schema for the readers interested in a specific problem context. We provide the following broad classification:-

Manufacturing, Production and Inventory Planning

1) Multi-item capacitated lot sizing problem. (Application 3.10)

2) Allocation of inspection effort on a production line. (Application 3.11)

3) Repair kit selection problem with limits on annual inventory budget. (Application 4.4)

4) Repair kit selection problem subject to a budget constraint. (Application 4.5)

5) Operations sequencing in discrete parts manufacturing. (Application 4.1)

Scheduling

1) Crew scheduling in a mass transit setting. (Application 3.3)

2) Rotating workforce scheduling. (Application 3.6)

3) Rotating workforce scheduling : additional applications. (Application 3.7)

4) Scheduling of power generation systems. (Application 3.9)

5) Multi-depot vehicle scheduling problem. (Application 6.3)

6) Scheduling daily operations in a steel rolling mill. (Application 6.5)

Social Sciences, Public Policy and Management Science

1) Rostering problem. (Application 3.4)

2) Capital budgeting with side constraints. (Application 4.2)

3) Capital budgeting with threshhold constraint. (Application 4.3)

4) Forest scheduling problem. (Application 4.7)

5) Manpower planning problem with attrition constraint. (Application 5.2)

6) Manpower planning problem with budget constraint. (Application 5.3)

7) Personnel assignment in armed forces. (Application 6.6)

Computer Science and Communication Systems

1) VLSI circuit design. (Application 3.1)

2) Survivable network design testing. (Application 3.2)

3) Multilocation facility modernization. (Application 5.1)

4) Concentrator location problem. (Application 7.5)

5) Routing electronic data. (Application 7.7)

6) Routing for circuit data networks. (Application 7.10)

Distribution and Transportation

1) Selecting freight handling terminals. (Application 4.6)

2) Routing of container ships. (Application 5.4)

3) Designing optimal railroad operating plan. (Application 7.1)

4) Delivery problem. (Application 7.2)

5) Routing in point-to-point delivery system. (Application 7.6)

Applied Mathematics and Theoretical Applications

1) Resource constrained shortest path problem. (Application 3.5)

2) Travelling salesman problem (TSP). (Application 3.8)

3) Network flow problems with variable upper bounds. (Application 5.5)

4) Multicommodity flow problem. (Application 5.6)

5) Set partitioning problem. (Application 5.7)

6) Equal flow problem. (Application 5.8)

7) Node-weighted Steiner tree problem. (Application 5.9)

8) Lagrangian approach to TSP. (Application 6.1)

9) Maximum collection problem. (Application 6.2)

10) Prize collecting TSP. (Application 6.4)

11) Capacitated fixed-charge network flow problem. (Application 7.3)

12) Steiner problem. (Application 7.4)

13) Constrained arborescence problem. (Application 7.8)

14) Minimal spanning tree with budget constraint. (Application 7.9)

In this classification, we have placed each application in a single category, thus reflecting the application's primary context. It is possible that several of these applications may fit into several of the categories. Our classification, however, does not depict any such overlaps.

This report serves several purposes. For the first time, it collects all such applications that are network flow problems with additional side contraints. These applications demonstrate the use of Lagrangian relaxation technique to solve such hard problems. As mentioned in the introduction given in Chapter 1, this work is in continuation of a project done last year on the same topic. Therefore, there is a similarity in the approach adopted in both the works; however, the applications searched by both are non-overlapping. A synthesis of both the works, therefore, will

provide over 80 different applications in varied contexts. Thus, such a work shall prove to be of great use to teachers who may include some of these in their courses on network/combinatorial optimization. We believe that the appreciation of the models described in this work will motivate the researchers to identify more and more hard optimization problems that can be solved by efficient network flow algorithms using Lagrangian relaxation. Additionally, as a direction for future research, a computational investigation strategy is being contemplated that compares the Lagrangian relaxation based solution technique with other techniques available to solve constrained network flow problems. Any such solution technique, Lagrangian relaxation based or otherwise, may be tested on a variety of constrained network flow problems (randomly generated as well as derived from real-life situation) and with varying number of constraints. The results obtained from such a study shall validate the use of Lagrangian relaxation as a strong algorithmic tool for solving hard optimization problems.

REFERENCES

Agrawal, V. 1985. A Lagrangian Relaxation Method for the Constrained Assignment Problem. *Computers and Operations Research* **12,** 97-106.

Ahuja, R.K., T.L. Magnanti, and J.B. Orlin. 1989. Network Flows: Handbook in Operations Research and Management Science. **Vol. 1:** *Optimization*. (Eds.) G.L. Nemhauser et al., 211-369.

Ahuja, R.K., T.L. Magnanti, and J.B. Orlin. 1993. *Network Flows: Theory, Algorithms and Applications*. Prentince Hall Inc., Englewood Cliffs, NJ, USA.

Ali, A.I., J. Kennington, and B. Shetty. 1988. The Equal Flow Problem. *European Journal of Opeartions Research* **36,** 107-115.

Ali, A.I., and H. Thiagarajan. 1989. A Network Relaxation Based Enumeration Algorithm for Set Partitioning. *European Journal of Opeartions Research* **38,** 76-85.

Altinkemer, K., and B. Gavish. 1991. Parallel Saving Based Heuristic for the Delivery Problem. *Operations Research* **39,** 456-469.

Balakrishnan, N., and R.T. Wong. 1990. A Network Model for the Rotating Workforce Scheduling Problem. *Networks* **20,** 25-42.

Balas, E. 1989. The Prize Collecting Travelling Salesman Problem. *Networks* **19,** 621-636.

Balas, E., and N. Christofides. 1981. A Restricted Lagrangean Approach to Travelling Salesman Problem. *Mathematical Programming* **21,** 19-46.

Bard, J.F., and T. A. Feo. 1989. Operations Sequencing in Discrete Parts Manufacturing. *Management Science* **35,** 249-255.

Beasley, J.E. 1984. An Algorithm for the Steiner Problem in Graphs. *Networks* **14,** 147-159.

Beasley, J.E., and N. Christofides 1989. An Algorithm for the Resource Constrained Shortest Path Problem. *Networks* **19,** 379-384.

Belling-Seb, K.P. Mevert, and C. Muller. 1988. Network Flow Problems With One Side Constraint: A Comparision of Three Solution Methods. *Computers and Operations Research* **15**, 381-394.

Cararessi, P., and G. Gallo. 1984. Network Models for Vehicle and Crew Scheduling. *Operational Research* **16**, 139-151.

Chen, W. H., and J. M. Thizy. 1990. Analysis of Relaxations for the Multi-item Capacitated Lot Sizing Problem. *Annals of Operations Research* **26**, 29-72.

Everett, H. 1963. Generalized Lagrange Multiplier Method for Solving Problems of Optimal Allocation of Resources. *Operations Research* **11**, 399-417.

Feo, T.A., and D. S. Hochbaum. 1986. Lagrangian Relaxation for Testing Infeasibility in VLSI Routing. *Operations Research* **34**, 819-831.

Fisher, M.L. 1981. The Lagrangian Relaxation Methods for Solving Integer Programming Problems. *Management Science* **27**, 1-18.

Geoffrion, A.M. 1974. Lagrangian Relaxation for Integer Programming. *Mathematical Programming Study* **2**, 82-114.

Guignard, M., and M.B. Rosenwein. 1989. An Application-Oriented Guide for Designing Lagrangean Dual Ascent Algorithms. *European Journal of Operations Research* **43**, 197-205.

Guignard, M., and M.B. Rosenwein. 1990. An Application of Lagrangean Decomposition to Resource - Constrained Minimum Weighted Arborescence Problem. *Networks* **20**, 345-359.

Hausman, H. 1978. Integer Programming and Related Areas: A Classified Bibliography. *Lecture Notes in Economics and Mathematical Systems.* **Vol.160.** Springer-Verlag, Berlin.

Held, M., and R. M. Karp. 1970. The Travelling Salesman Problem and Minimum Spanning Trees. *Operations Research* **18**, 1138-1162.

Held, M., and R. M. Karp. 1971. The Travelling Salesman Problem and Minimum Spanning Trees. *Mathematical Programming* **1**, 6-25.

Held, M., P. Wolf, and H. D. Crowder. 1974. Validation of Subgradient Optimization. *Mathematical Programming* **6,** 62-88.

Hochbaum, D. S., and A. Segev. 1989. Analysis of Flow Problems With Fixed Charges. *Networks* **19,** 291-312.

Jornsten, K., and S. Migdalas. 1988. Designing a Minimal Spanning Tree Subject to a Budget Constraint. *Optimization* **19,** 475-484.

Kasting, C. 1976. Integer Programming and Related Areas: A Classified Bibliography. *Lecture Notes in Economics and Mathematical Systems.* **Vol.128.** Springer-Verlag, Berlin.

Kataoka, S., and S. Morito. 1988. An Algorithm for Single Constraint Maximum Collection Problem. *Journal of the Operations Research Society of Japan* **31,** 515-530.

Keaton, M. H. 1989. Designing Optimal Railroad Operating Plans: Lagrangian Relaxation and Heuristic Approaches. *Transportation Research* **23B,** 415-431.

Khang, D.B., and O. Fujiwara. 1991. Approximate Solutions of Capacitated Fixed-Charge Minimum Cost Network Flow Problems. *Networks* **21,** 689-704.

Leung, J.M.Y., T.L. Magnanti, and V. Singhal. 1990. Routing in Point-to-Point Delivery Systems: Formulations and Solutios Heuristics. *Transportation Science* **24,** 245-260.

Magnanti, T.L., and R.T. Wong. 1984. Network Design and Transportation Planning: Models and Algorithms. *Transportation Science* **18,** 1-53.

Mamer, J. W., and A. W. Shogan. 1987. A Constrained Capital Budgeting Problem With Applications to Repair Kit Selection. *Management Science* **33,** 800-806.

Mason, L.G., A. Girard, and H.D. Gu. 1990. Multilocation Facility Modernization: Models and Heuristics. *Operations Research* **38,** 412-425.

Mirzaian, A. 1985. Lagrangian Relaxation for the Star-Star Concentrator Location Problem: Approximation Algorithm and Bounds. *Networks* **15,** 1-20.

Muckstadt, J. A., and S. A. Koenig. 1977. An Application of Lagrangian Relaxation to Scheduling in Power - Generation Systems. *Operations Research* **25**, 387-403.

Orlin, J. B., and D. Bertsimas. 1991. A Technique for Speeding up the Solution of the Lagrangean Dual. Working Paper 3728-91-MSA, Sloan School of Management, MIT.

Price, W. L. 1978. Solving Goal-Programming Manpower Models Using Advanced Network Codes. *Journal of Operations Research Society* **29**, 1231-1239.

Price, W. L., and M. Gravel. 1984. Solving Network Manpower Problems With Side Constraints. *European Journal of Operations Research* **15**, 196-202.

Rana, K., and R. G. Vickson. 1991. Routing Container Ships using Lagrangean Relaxation and Decomposition. *Transportation Science* **25**, 201-214.

Saviozzi, G. 1986. Advanced Start for the Multicommodity Network Flow Problem. *Mathematical Programming Study* **26**, 221-224.

Segev, A. 1987. The Node-Weighted Steiner Tree Problem. *Networks* **17**, 1-17.

Shapiro, J.F. 1991. Convergent Duality for the Travelling Salesman Problem. *Operations Research Letters* **10**, 129-136.

Shetty, B. 1990. A Heuristic Algorithm for a Network Problem With Variable Upper Bounds. *Networks* **20**, 373-389.

Yee, J. R., and F.Y.S. Lin. 1992. A Routing Algorithm for Virtual Circuit Data Networks With Multiple Sessions per O-D Pair. *Networks* **22**, 185-208.